HEWLETT PACKARD

HP 3000 SERIES II
COMPUTER SYSTEM

SYSTEM MICROPROGRAM LISTING

Manual Part No. 30000-90023
Microfiche Part No. 30000-90037

Printed in U.S.A. 8/76

HEWLETT-PACKARD
5303 STEVENS CREEK BLVD., SANTA CLARA, CALIFORNIA, 95050

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the most recent date on which the technical material on any given page was altered. If a page is simply re-arranged due to a technical change on a previous page, it is not listed as a changed page. Within the manual, changes are marked with a vertical bar in the margin.

PRINTING HISTORY

New editions incorporate all update material since the previous edition. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date on the title page and back cover changes only when a new edition is published. If minor corrections and updates are incorporated, the manual is reprinted but neither the date on the title page and back cover nor the edition change.

First edition	June 1, 1976	
Update 1	June 15, 1976	Revised to incorporate Revision A changes and to add Extended Instruction Set Listing
Second edition	August 13, 1976	Revised to incorporate Update 1 and Revision B Microprogram Listing

PREFACE

This document consists of three elements.

1. The Look-up Table. The table is located following the first divider.
2. The HP 3000 Series II Microprogram Listing. The listing follows the second divider.
3. The HP 3000 Series II Extended Instruction Set Listing. The Listing follows the third divider.
4. The Microprogramming Language Description. The description follows the fourth divider.

HP 3000 Series II

Computer System

LOOK
UP
TABLE

6-15-76

```
000  1111 1111 7777          PARITY (SEE PART LISTINGS FOR ACTUAL DATA)
001  1111 1111 7777          UNNASSIGNED
002  1111 1111 7777
003  1111 1111 7777
004  1111 1111 7777
005  1111 1111 7777
006  1111 1111 7777
007  1111 1111 7777
010  1111 1111 7777
011  1111 1111 7777
012  1111 1111 7777
013  1111 1111 7777
014  1111 1111 7777
015  1111 1111 7777
016  1111 1111 7777
017  1111 1111 7777
                     ************************************************

                     04 - 17  MEMORY REFERENCE INSTRUCTIONS
                     4 ENTRIES PER OPCODE, NO STACK PREADJUST ALLOWED.
                     W = 1-INSURES CORRECT JLUI MAPPING.
                     PADD IS CORRECT DISPLACEMENT IN ALL INSTRUCTIONS.
                     X IS INCLUDED IF INDEXED, NOT INDIRECT.
                     AUTOMATIC X/2 FOR LDB/STB PADD IF REQUIRED.
                     AUTOMATIC P*X FOR LDD/STD PADD IF REQUIRED.
                     PADD, BASE FORCED ON FIRST LINE OF MICROCODE
                     DURING NEXT+1 CYCLE.

                     ************************************************

020  1111 1110 0043  AC1D  04   DBQ    LOAD   0    8 - 15   1  YES
021  1111 1010 0042  AC1S       S-            0   10 - 15   1  YES
022  1111 1111 0101  LOAD       JLUI          0              1  NO    MICROCODE INSURES
                                                                      SR < 4
023  1111 1110 0060  AC1P       P             0    8 - 15   1  YES
024  1111 1110 0154  AC4D  05   DBQ    STOR   0    8 - 15   1  YES
```

LUT	CONTROL	RAR	LABEL	OP CODE	ENTRY	INSTR	SR>=	PREADDER	W	JLUI	COMMENTS
025	1111 1010	0153	AC4S		S-		0	10 - 15	1	YES	
026	1111 1111	0211	STOR		JLUI		0		1	NO	MICROCODE INSURES SR > 0
027	1111 1111	0502	MTBI		P	LOOP C	0	8 - 15	1	NO	INSURE SR > 2 (TBX,MTBX) OR SR = 3 (TBA,MTBA)
030	1111 1110	0043	AC1D	06	DBQ	CMPM	0	8 - 15	1	YES	
031	1111 1010	0042	AC1S		S-		0	10 - 15	1	YES	
032	1111 1111	0004	CMPM		JLUI		0		1	NO	MICROCODE INSURES SR > 0
033	1111 1110	0060	AC1P		P		0	8 - 15	1	YES	
034	1111 1110	0043	AC1D	07	DBQ	ADDM	0	8 - 15	1	YES	
035	1111 1010	0042	AC1S		S-		0	10 - 15	1	YES	
036	1111 1111	0075	ADDM		JLUI		0		1	NO	MICROCODE INSURES SR > 0
037	1111 1110	0060	AC1P		P		0	8 - 15	1	YES	
040	1111 1110	0043	AC1D	10	DBQ	SUBM	0	8 - 15	1	YES	
041	1111 1010	0042	AC1S		S-		0	10 - 15	1	YES	
042	1111 1111	0075	ADDM		JLUI		0		1	NO	MICROCODE INSURES SR > 0
043	1111 1110	0060	AC1P		P		0	8 - 15	1	YES	
044	1111 1110	0043	AC1D	11	DBQ	MPYM	0	8 - 15	1	YES	
045	1111 1010	0042	AC1S		S-		0	10 - 15	1	YES	
046	1111 1111	0702	MPYM		JLUI		0		1	NO	MICROCODE INSURES SR > 0
047	1111 1110	0060	AC1P		P		0	8 - 15	1	YES	
050	1111 1110	0043	AC1D	12	DBQ	DECM	0	8 - 15	1	YES	
051	1111 1010	0042	AC1S		S-		0	10 - 15	1	YES	
052	1111 1111	0011	IDMY		JLUI		0		1	NO	
053	1111 1110	0070	AINC		DQS	INCM	0	8 - 15	1	YES	MICROCODE SPLITS DBQ/S-. PADD IS CORRECT (DB/Q/S)
054	1111 1110	0043	AC1D	13	DBQ	LDX	0	8 - 15	1	YES	
055	1111 1010	0042	AC1S		S-		0	10 - 15	1	YES	
056	1111 1111	0021	LDX		JLUI		0		1	NO	
057	1111 1110	0060	AC1P		P		0	8 - 15	1	YES	
060	1110 1111	0401	BRD	14	DBQ	BR/BCC	0	8 - 15	1	NO	Z = 0 MICROCODE SPLITS BR/BCC. PADD IS CORRECT
061	1110 1011	0400	BRS		S-		0	10 - 15	1	NO	Z = 0 MICROCODE SPLITS BR/BCC. PADD IS CORRECT
062	1111 1111	7777			JLUI		0		1	NO	NOT USED
063	1111 1111	0412	BRP		P	BR	0	8 - 15	1	NO	
064	1111 1110	0126	AC3D	15	DBQ	LDD	0	8 - 15	1	YES	
065	1111 1010	0125	AC3S		S-	LDD	0	10 - 15	1	YES	
066	1111 1111	0142	LDD		JLUI		0		1	NO	MICROCODE INSURES SR < 3
067	1111 1111	0222	ALSB		DQS	LDB	0	8 - 15	1	NO	MICROCODE SPLITS DBQ/S-. PADD IS CORRECT (DB/Q/S)
070	1111 1110	0166	AC5D	16	DBQ	STD	0	8 - 15	1	YES	
071	1111 1010	0165	AC5S		S-	STD	0	10 - 15	1	YES	
072	1111 1111	0200	STD		JLUI		0		1	NO	

LUT	CONTROL	RAR	LABEL	OP CODE	ENTRY	INSTR	SR>=	PREADDER	W	JLUI	COMMENTS
073	1111 1111	0222	ALSB		DQS	STB	0	8 - 15	1	NO	MICROCODE SPLITS DBQ/S-, PADD IS CORRECT (DB/Q/S)
074	1111 1110	0104	AC2D	17	DBQ	LRA	0	8 - 15	1	YES	
075	1111 1010	0103	AC2S		S-		0	10 - 15	1	YES	
076	1111 1111	0123	LRA		JLUI		0		1	NO	MICROCODE INSURES SR < 4
077	1111 1110	0114	AC2P		P		0	8 - 15	1	YES	

****

SUBOP 1 SHIFTS AND BRANCHES
SINGLE ENTRY PER OPCODE.
PADD = SHIFT COUNT FOR ALL SHIFTS, WITH W = 0
INHIBITING BIT 10 SIGN INTERPRETATION.
PADD IS NOT INDEXED; MICROCODE USES XC OPTION.
CTSS, CTSD DETERMINE TYPE OF SHIFT FOR SHARED
SHIFT MICROCODE ENTRIES.
FOR BRANCHES, PADD = P REL. DISPLACEMENT, W = 1
ENABLES SIGN BIT, NO INDEXING.

****

100	1101 1001	1263	SHFL	SUBOP 1 = 00	ASL	1	10 - 15	0	NO	
101	1101 1001	1255	SHFR		ASR	1	10 - 15	0	NO	
102	1101 1001	1263	SHFL		LSL	1	10 - 15	0	NO	
103	1101 1001	1255	SHFR		LSR	1	10 - 15	0	NO	
104	1101 1001	1263	SHFL		CSL	1	10 - 15	0	NO	
105	1101 1001	1255	SHFR		CSR	1	10 - 15	0	NO	
106	1101 1001	1422	SCAN		SCAN	1	10 - 15	0	NO	
107	1101 0111	0456	IABZ		IABZ	1	11 - 15	1	NO	
110	1001 1001	1304	TASL		TASL	3	10 - 15	0	NO	
111	1001 1001	1317	TASR		TASR	3	10 - 15	0	NO	
112	1111 0111	0450	IXBZ		IXBZ	0	11 - 15	1	NO	
113	1111 0111	0453	DXBZ		DXBZ	0	11 - 15	1	NO	
114	1111 0111	0445	BCY		BCY	0	11 - 15	1	NO	
115	1111 0111	0443	BNCY		BNCY	0	11 - 15	1	NO	
116	1001 1001	1323	TNSL		TNSL	3	10 - 15	0	NO	
117	0111 1001	1336	QALR		QASL/QASR	4	10 - 15	0	NO	
120	1011 1001	1270	SHDL		DASL	2	10 - 15	0	NO	
121	1011 1001	1276	SHDR		DASR	2	10 - 15	0	NO	
122	1011 1001	1270	SHDL		DLSL	2	10 - 15	0	NO	
123	1011 1001	1276	SHDR		DLSR	2	10 - 15	0	NO	
124	1011 1001	1270	SHDL		DCSL	2	10 - 15	0	NO	
125	1011 1001	1276	SHDR		DCSR	2	10 - 15	0	NO	
126	1011 0111	0472	CPRB		CPRB	2	11 - 15	1	NO	
127	1101 0111	0461	DABZ		DABZ	1	11 - 15	1	NO	
130	1111 0111	0434	BOV		BOV	0	11 - 15	1	NO	
131	1111 0111	0440	BNOV		BNOV	0	11 - 15	1	NO	
132	1101 1001	1440	TBC		TBC	1	10 - 15	0	NO	
133	1101 1001	1432	TRBC		TRBC	1	10 - 15	0	NO	
134	1101 1001	1434	TSBC		TSBC	1	10 - 15	0	NO	
135	1101 1001	1436	TCBC		TCBC	1	10 - 15	0	NO	
136	1101 0111	0467	BRO		BRO	1	11 - 15	1	NO	
137	1101 0111	0464	BRE	SUBOP 1 = 37	BRE	1	11 - 15	1	NO	

```
                        **************************************************

                        SUBOP 2  MOVES, MINIS
                        DOUBLE ENTRY PER MOVEOP, SINGLE ENTRY PER MINIOP,
                        W = 0/1 SPECIFIES +/- PADD.

                        **************************************************

140  1001 1101 2001   MVWP   SUB 2 MOVEOPS   MOVF PB   3     8 - 15   0  NO
141  1001 1101 2000   MVWD                    MOVE DB   3     8 - 15   0  NO
142  1001 1101 2047   MVBP                    MVB  PB   3     8 - 15   0  NO
143  1001 1101 2046   MVBD                    MVB  DB   3     8 - 15   0  NO
144  0111 0001 2266   MABS                    MABS/MVBL 4    12 - 15   0  NO   MICROCODE CHECKS
                                                                                 CIR(13)
145  1011 0001 2162   SCW                     SCW/MTDS  2    12 - 15   0  NO
146  0111 0001 2267   MDS                     MDS/MVLR  4    12 - 15   0  NO   MICROCODE CHECKS
                                                                                 CIR(13)
147  1011 0001 2161   SCU                     SCU/MFDS  2    12 - 15   0  NO
150  1011 1101 2025   MVBW                    MVBW -N   2     8 - 15   0  NO
151  1011 1101 2025   MVBW                    MVBW  N   2     8 - 15   0  NO
152  1001 1101 2047   MVBP                    CMPB PB   3     8 - 15   0  NO
153  1001 1101 2046   MVBD                    CMPB DB   3     8 - 15   0  NO
154  0111 1101 1570   LLSH   SUB 2 MINIOPS .  RSW/LLSH  4     8 - 15   0  NO   MICROCODE CHECKS
                                                                                 CIR(15)
155  1111 1101 0307   PLSA                    PLDA/PSTA 0     8 - 15   0  NO   MICROCODE CHECKS
                                                                                 CIR(15)
156  1011 1101 0323   LSAB                    EXT ADDR  2     8 - 15   0  NO   MICROCODE CHECKS
                                                                                 CIR(14-15)
157  1111 0001 2535   IXIT                    PROCESS   0    12 - 15   0  NO   IXIT, LOCK, PCN,
                                                                                 UNLK DECODED FROM
                                                                                 PADD(14-15)

160  1111 1111 7777         UNNASSIGNED
```

```
                        ************************************************

                        SUBOP 2 = 00/17
                        SINGLE ENTRY PER INSTRUCTION.
                        PADD IS CIR(8-15), W = 0/1 SPECIFIES +/- PADD.

                        ************************************************

161  0111 1101 1613   OPTX  SUBOP 2 = 01   OPTIONS   4      8 - 15   0  NO
162  1111 1101 0751   LDI                  LDI       0      8 - 15   0  NO
163  1111 1101 0753   LDXI                 LDXI      0      8 - 15   0  NO
164  1101 1111 0537   CMPI                 CMPI      1      8 - 15   1  NO    -PADD
165  1101 1101 0760   ADDI                 ADDI      1      8 - 15   0  NO
166  1101 1111 0760   ADDI                 SUBI      1      8 - 15   1  NO    -PADD
167  1101 1101 0701   MPYI                 MPYI      1      8 - 15   0  NO
170  1101 1101 0542   DIVI                 DIVI      1      8 - 15   0  NO
171  1111 1101 1446   PSHR                 PSHR      0      8 - 15   0  NO

172  1111 1111 0751   LDI                  LDNI      0      8 - 15   1  NO    -PADD
173  1111 1111 0753   LDXI                 LDXN      0      8 - 15   1  NO    -PADD
174  1101 1101 0537   CMPI                 CMPN      1      8 - 15   0  NO
175  1101 1101 1400   DEXF                 EXF       1      8 - 15   0  NO
176  1011 1101 1400   DEXF                 DPF       2      8 - 15   0  NO
177  0111 1101 1476   SETR  SUBOP 2 = 17   SETR      4      8 - 15   0  NO
```

```
                    ************************************************

                    SPEC 3 = 00/17
                    SINGLE ENTRY PER INSTRUCTION.
                    PADD IS CIR(12-15), W = 0/1 SPECIFIES +/- PADD.

                    ************************************************

200  1101 0001 0347  LST   SPEC 3 = 00     LST        1     12 - 15   0  NO
201  1110 0001 2764  PAUS                  PAUS       0     12 - 15   0  NO   Z = 0
202  1111 0001 1674  SED                   SED        0     12 - 15   0  NO
203  1011 0001 1544  XCHD                  XCHD/MPE   2     12 - 15   0  NO   XCHD, PSDB, DISP,
                                                                               PSEP DECODED FROM
                                                                               PADD(14-15)
204  1101 0011 1706  SMSK                  SMSK/SCLK  1     12 - 15   1  NO   MICROCODE CHECKS
                                                                               CIR(15), -PADD
205  1111 0011 1701  RMSK                  RMSK/RCLK  0     12 - 15   1  NO   MICROCODE CHECKS
                                                                               CIR(15), -PADD
206  1100 0001 1561  XEQ                   XEQ        1     12 - 15   0  NO   Z = 0
207  1101 0001 1617  SIO                   SIO        1     12 - 15   0  NO
210  1111 0001 1630  RIO                   RIO        0     12 - 15   0  NO
211  1101 0001 1641  WIO                   WIO        1     12 - 15   0  NO
212  1111 0001 1653  TIO                   TIO        0     12 - 15   0  NO
213  1101 0001 1660  CIO                   CIO        1     12 - 15   0  NO
214  1101 0001 1670  CMD                   CMD        1     12 - 15   0  NO
215  1011 0001 0360  SST                   SST        2     12 - 15   0  NO
216  1111 0001 1664  SIN                   SIN        0     12 - 15   0  NO
217  1111 0001 2757  HALT  SPEC 3 = 17     HALT       0     12 - 15   0  NO
220  1111 1111 7777        UNNASSIGNED                                  NO
```

```
                    **************************************************

                    SUBOP 3 = 01/17
                    SINGLE ENTRY PER INSTRUCTION
                    PADD IS CIR(8-15), W = 0/1 SPECIFIES +/- PADD.

                    **************************************************

221  1111 1101 2411   SCAL  SUBOP 3 = 01    SCAL      0      8 - 15   0  NO
222  1111 1101 2412   PCAL                   PCAL      0      8 - 15   0  NO
223  1111 1101 2456   EXIT                   EXIT      0      8 - 15   0  NO
224  1101 1111 2444   SXIT                   SXIT      1      8 - 15   1  NO   -PADD
225  1111 1101 0754   ADXI                   ADXI      0      8 - 15   0  NO
226  1111 1111 0754   ADXI                   SBXI      0      8 - 15   1  NO   -PADD
227  1111 1101 2400   LLBL                   LLBL      0      8 - 15   0  NO
230  1111 1101 0277   LDPB                   LDPP      0      8 - 15   0  NO
231  1111 1111 0277   LDPB                   LDPN      0      8 - 15   1  NO   -PADD
232  1101 1101 1553   ADDS                   ADDS      1      8 - 15   0  NO
233  1101 1111 1552   SUBS                   SUBS      1      8 - 15   1  NO   -PADD
234  1111 1111 7777                          SPARE                         NO
235  1101 1101 0755   ORI                    ORI       1      8 - 15   0  NO
236  1101 1101 0756   XORI                   XORI      1      8 - 15   0  NO
237  1101 1101 0757   ANDI                   ANDI      1      8 - 15   0  NO
240  1111 1111 7777         UNASSIGNED
241  1111 1111 7777
242  1111 1111 7777
243  1111 1111 7777
244  1111 1111 7777
245  1111 1111 7777
246  1111 1111 7777
247  1111 1111 7777
250  1111 1111 7777
251  1111 1111 7777
252  1111 1111 7777
253  1111 1111 7777
254  1111 1111 7777
255  1111 1111 7777
256  1111 1111 7777
257  1111 1111 7777
260  1111 1111 7777
261  1111 1111 7777
262  1111 1111 7777
263  1111 1111 7777
264  1111 1111 7777
265  1111 1111 7777
266  1111 1111 7777
267  1111 1111 7777
270  1111 1111 7777
271  1111 1111 7777

272  1111 1111 7777
273  1111 1111 7777
274  1111 1111 7777
275  1111 1111 7777
276  1111 1111 7777
277  1111 1111 7777
```

```
                        ************************************************

                        STACKOPS
                        SINGLE ENTRY PER INSTRUCTION.
                        W, PADD ARE DON'T CARES.
                        DEFAULT IS PADD = -CIR(8-15).

                        ************************************************
```

300	1111	1111	0564	NOP	STACKOPS	NOP	0	8 - 15	1	NO
301	1011	1111	0646	DELB		DELB	2	8 - 15	1	NO
302	1011	1111	0644	DDEL		DDEL	2	8 - 15	1	NO
303	1111	1111	0775	ZROX		ZROX	0	8 - 15	1	NO
304	1111	1111	0552	INCX		INCX	0	8 - 15	1	NO
305	1111	1111	0553	DECX		DECX	0	8 - 15	1	NO
306	1111	1111	0773	ZERO		ZERO	0	8 - 15	1	NO
307	1111	1111	0767	DZRO		DZRO	0	8 - 15	1	NO
310	0111	1111	0634	DCMP		DCMP	4	8 - 15	1	NO
311	0111	1111	0620	DADD		DADD	4	8 - 15	1	NO
312	0111	1111	0624	DSUB		DSUB	4	8 - 15	1	NO
313	1011	1111	0705	MPYL		MPYL	2	8 - 15	1	NO
314	1001	1111	0724	DIVL		DIVL	3	8 - 15	1	NO
315	1011	1111	0630	DNEG		DNEG	2	8 - 15	1	NO
316	0111	1111	0574	DXCH		DXCH	4	8 - 15	1	NO
317	1011	1111	0615	CMP		CMP	2	8 - 15	1	NO
320	1011	1111	0612	ADD		ADD	2	8 - 15	1	NO
321	1011	1111	0613	SUB		SUB	2	8 - 15	1	NO
322	1011	1111	0704	MPY		MPY	2	8 - 15	1	NO
323	1011	1111	0721	DIV		DIV	2	8 - 15	1	NO
324	1101	1111	0614	NEG		NEG	1	8 - 15	1	NO
325	1101	1111	1261	TEST		TEST	1	8 - 15	1	NO
326	1011	1111	0610	STBX		STBX	2	8 - 15	1	NO
327	1011	1111	0560	DTST		DTST	2	8 - 15	1	NO
330	1011	1111	1212	DFLT		DFLT	2	8 - 15	1	NO
331	1101	1111	1262	BTST		BTST	1	8 - 15	1	NO
332	1011	1111	0572	XCH		XCH	2	8 - 15	1	NO
333	1101	1111	0554	INCA		INCA	1	8 - 15	1	NO
334	1101	1111	0555	DECA		DECA	1	8 - 15	1	NO
335	1101	1111	0566	XAX		XAX	1	8 - 15	1	NO
336	1101	1111	0605	ADAX		ADAX	1	8 - 15	1	NO
337	1101	1111	0603	ADXA		ADXA	1	8 - 15	1	NO

LUT	CONTROL	RAR	LABEL OP CODE ENTRY	INSTR	SR>=	PREADDER	W	JLUI	COMMENTS
340	1101 1111	0645	DEL	DEL	1	8 - 15	1	NO	
341	1011 1111	0640	ZROB	ZROB	2	8 - 15	1	NO	
342	1011 1111	0606	LDXB	LDXB	2	8 - 15	1	NO	
343	1101 1111	0604	STAX	STAX	1	8 - 15	1	NO	
344	1111 1111	0601	LDXA	LDXA	0	8 - 15	1	NO	
345	1101 1111	0761	DUP	DUP	1	8 - 15	1	NO	
346	1011 1111	0763	DDUP	DDUP	2	8 - 15	1	NO	
347	1101 1111	1207	FLT	FLT	1	8 - 15	1	NO	
350	0111 1111	1201	FCMP	FCMP	4	8 - 15	1	NO	
351	0111 1111	1001	FADD	FADD	4	8 - 15	1	NO	
352	0111 1111	1000	FSUB	FSUB	4	8 - 15	1	NO	
353	0111 1111	1060	FMPY	FMPY	4	8 - 15	1	NO	
354	0111 1111	1110	FDIV	FDIV	4	8 - 15	1	NO	
355	1011 1111	1175	FNEG	FNEG	2	8 - 15	1	NO	
356	1001 1111	0577	CAB	CAB	3	8 - 15	1	NO	
357	1011 1111	0642	LCMP	LCMP	2	8 - 15	1	NO	
360	1011 1111	0647	LADD	LADD	2	8 - 15	1	NO	
361	1011 1111	0650	LSUB	LSUB	2	8 - 15	1	NO	
362	1011 1111	0655	LMPY	LMPY	2	8 - 15	1	NO	
363	1001 1111	0663	LDIV	LDIV	3	8 - 15	1	NO	
364	1101 1111	0654	NOT	NOT	1	8 - 15	1	NO	
365	1011 1111	0023	OR	OR	2	8 - 15	1	NO	
366	1011 1111	0027	XOR	XOR	2	8 - 15	1	NO	
367	1011 1111	0033	AND	AND	2	8 - 15	1	NO	
370	1011 1111	1223	FIXR	FIXR	2	8 - 15	1	NO	
371	1011 1111	1222	FIXT	FIXT	2	8 - 15	1	NO	
372	1111 1111	7777		SPARE				NO	
373	1011 1111	0556	INCB	INCB	2	8 - 15	1	NO	
374	1011 1111	0557	DECB	DECB	2	8 - 15	1	NO	
375	1011 1111	0570	XBX	XBX	2	8 - 15	1	NO	
376	1011 1111	0611	ADBX	ADBX	2	8 - 15	1	NO	
377	1011 1111	0607	ADXB	ADXB	2	8 - 15	1	NO	

HP 3000 SERIES II

MICROPROGRAM LISTING

8-13-76

```
1                       *     MC-3000/II REV. B  (2K PARITY)
2                       *     SHOULD NOT HAVE PARITY FOR 6.4-7 SINCE RAR TARGETS MAY BE ADDED.
3                       *       (6.0-7 DO NOT HAVE PARITY)
4                       *     SHOULD NOT HAVE PARITY FOR 14 AND 15 SINCE MORE CODE MAY BE ADDED.
5                       *
6                       *
7                       *
8                       *   *******   ********  ********  ********        **    ********  ********
9                       *   *******   ********  ********  ********        **    ********  ********
10                      *         **  **     **  **     **  **     **      **      **        **
11                      *         **  **     **  **     **  **     **      **      **        **
12                      *    ******   **     **  **     **  **     **     **      **        **
13                      *    ******   **     **  **     **  **     **     **      **        **
14                      *         **  **     **  **     **  **     **    **      **        **
15                      *         **  **     **  **     **  **     **    **      **        **
16                      *   *******   ********  ********  *******.**    **      ********  ********
17                      *   *******   ********  ********  *******.**    **      ********  ********
18                      *
19                      *
20                      *
21                      *
22                      *
23                      *
24                      *
25                      *
26                      *   *      *  ********  ********  ********  ********
27                      *   **     **  ********  ********  ********  ********
28                      *   ***   ***     **     **        **     **  **     **
29                      *   ********      **     **        **     **  **     **
30                      *   ** ** **      **     **        ********  **     **
31                      *   **    **      **     **        *******.  **     **    ******
32                      *   **    **      **     **        **  **    **     **    ******
33                      *   **    **      **     **        **  **    **     **
34                      *   **    **      ********  ********  **    **    ********
35                      *   **    **      ********  ********  **    **    ********
36                      *
37                      *
38                      *
39                      *
40                      *
41                      *
42                      *
43                      *
44                      *   *******   ********  ********  ********  ********      **      *      *
45                      *   *******   ********  ********  ********  ********    ****    **     **
46                      *   **    **  **    **  **    **  **    **  **    **    **  **   ***   ***
47                      *   **    **  **    **  **    **  **        **    **   **    **  *********
48                      *   *******   ********  **    **  ** ***.  ********  ********  ** ** **
49                      *   *******   ********  **    **  ** ***.  ********  ********  **    **
50                      *   **        **  **    **    **  ** ** **  **  **    **    **  **    **
51                      *   **        **   **   **    **  **    **  **   **   **    **  **    **
52                      *   **        **    **  ********  *******.  **    **  **    **  **    **
53                      *   **        **     ** ********  ********  **     **  **    **  **    **
54                      *
55                      *
```

```
                                *
                                *
                                *              SECTOR 0
                                *
                                *         PON, CPU RESET, IR HARDWARE ENTRIES
                                *
                                *         SR PREADJUST ROUTINE
                                *
                                *         BOOLEAN STACKOPS: OR, XOR, AND
                                *
                                *         ADDR COMPUTATION ROUTINES AC1=2S,D,P, AC3=5S,D, AINC, ALSB
                                *         FOR MEM REF INSTRS.  E REFERS TO EFFECTIVE OPERAND ADDR.
                                *
                                *         SUBROUTINE: TSCK
                                *
                                *         MEM REF INSTRS: ADDM, CMPM, DECM, INCM, LDB , LDD , LDPN,
                                *                         LDPP, LDX , LOAD, LRA , STB , STD , STOR,
                                *                         SUBM        (MPYM IS IN SECTOR 1)
                                *
                                *         PLDA/PSTA, LSEA/LDEA/SSEA/SDEA, LST, SST
                                *
                                *
 0000  37306303124                      JMP  TRP7 SP1       UNC       UNIM INSTR (WRAP AROUND)
 0001  37766302742                      JMP  PWR            UNC       POWER ON ENTRY
 0002  37766302765                      JMP  CPRS           UNC       CPU RESET ENTRY
 0003  37766303001                      JMP  IR             UNC       INTERRUPT ENTRY
                                *
                                *
                                *         CMPM
                                *    DB,Q,S OR P REL ADDR; ENTER WITH OPND=(E)
                                *
 0004  26317617777  CMPM        OPND ADD       SP1         SRNZ      SP1 - (E)
 0005  33762301732              RA   JSB  PUL1              UNC       FILL A TOS REG IF NEC
 0006  01767477556         UBUS SP1  SUB             POPA NOFL      CCA ON (S)-(E)
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
 0007  00773357753              RA   CIR  IOR             CCA  NEXT      REVERSE CCG, CCL IF OVFL
 0010  37777757777                   ADD                  NEXT        (CIR>0 SO NO CCE IF OVFL)
                                *
                                *
                                *         INCM, DECM
                                *    DB,Q OR S REL ADDR; ENTER WITH SP0=E, OPND=(E),
                                *    F2=INCM, F1 SET AND SP1=SM+SR-E IF E IS IN TOS REGS.
                                *
 0011  26326140015  IDM1        OPND JMP  IDM2 SP0         F1        IF F1 (E) IS FROM TOS REGS
 0012  37177567555              SP0       ADD       BUS  WRD  F2        (E) IS FROM MEM
 0013  37165357435              SP0       CADO      BUS  DATA NEXT      DECR MEM IF NF2
 0014  37174757435              SP0       INCO      BUS  DATA NEXT      ELSE INCR MEM
                                *
 0015  37074567775  IDM2        SP0       INCO      MREG      F2        INCR MEM
 0016  37065357775              SP0       CADO      MREG      NEXT      ELSE IF NF2 DECR MEM
 0017  37777757777                        ADD                 NEXT
                                *
                                *
                                *    SRP IS THE STACK PREADJUST ROUTINE; ENTERED AUTOMATICALLY
                                *      (AT 20,24,30,34 FOR PULL 1 TO 4) IF SR < THE VALUE SPECIFIED
```

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:05 PM

                        *      IN THE LUT FOR THE CURRENT INSTR.
                        *    EXIT WITH RANK1 JLUI WITH SM_SM-1.
                        *    LDX, OR, XOR, AND INSTRS ARE EMBEDDED IN HOLES IN SRP.
                        *
                        $00=0
0020    23137757777     SRP4       SM   ADD        BSP0 ROS  UNC       READ (SM); SKIP HOLE
                        *
                        *         LDX
                        *    DB,Q,S OR P REL ADDR; ENTER WITH OPND=(E)
0021    26557757757     LDX        OPND ADD        X    CCA  NEXT      LOAD X, NEXT
                        *
0022    37467367475        SP0        CAD        SM   SF1  UNC       DECR SM; SKIP HOLE
                        *
                        *         OR
                        *    ENTER WITH SR>=2                                (S-1)_(S)OR(S-1),
0023    32653357553     OR   RA   RB   IOR        RB   POPA NEXT        CCA, S_S-1, NEXT
                        *
0024    23137557777     SRP3       SM   ADD        BSP0 ROS  NF1       READ (SM), 1ST ENTRY IF NF1
0025    26277777217                OPND ADD        QUP  INSR           NO, QUP WORD
0026    37467367475        SP0        CAD        SM   SF1  UNC       DECR SM; SKIP HOLE
                        *
                        *         XOR
                        *    ENTER WITH SR>=2                                (S-1)_(S)XOR(S-1),
0027    32643357553     XOR  RA   RB   XOR        RB   POPA NEXT        CCA, S_S-1, NEXT
                        *
0030    23137557777     SRP2       SM   ADD        BSP0 ROS  NF1       READ (SM), IST ENTRY IF NF1
0031    26277777217                OPND ADD        QUP  INSR           NO, QUE WORD
0032    37467367475        SP0        CAD        SM   SF1  UNC       DECR SM; SKIP HOLE
                        *
                        *         AND
                        *    ENTER WITH SR>=2                                (S-1)_(S)AND(S-1),
0033    32643757553     AND  RA   RB   AND        RB   POPA NEXT        CCA, S_S-1, NEXT
                        *
0034    23137557777     SRP1       SM   ADD        BSP0 ROS  NF1       READ (SM), IST ENTRY IF NF1
0035    26277777217                OPND ADD        QUP  INSR           NO, QUP WORD
0036    22767127355        SP0   DB   CAD                  CLIB POS    CLIB; SM<=DB OR WRAP AROUND
0037    37766263120                   JMP  STUN                NPRV     YES, STUN IF NPRV
0040    26277777217                OPND ADD        QUP  INSR           QUP WORD, INSR
0041    37467317455        SP0        CAD        SM   CF1  JLUI      CF1, DECR SM, EX INSTR
                        *
                        *
                        *    AC1S,D IS THE S AND DB,Q REL ADDR COMPUTATION ROUTINE
                        *      FOR ADDM,CMPM,DECM,INCM,LDX,LOAD,MPYM,SUBM.
                        *    IF ENTERED AT AC1S SBUS=SM- RBUS=PADD+(XC IF NOT INDR).
                        *    IF ENTERED AT AC1D SBUS=DB+ OR Q+- RBUS=PADD+(XC IF NOT INDR)
                        *      INSTEAD OF SR+UBUS.
                        *    INCM ENTERS AT AC12 FROM AINC WITH F2 AND A RNK1 JMP WITH SP0_ADDR.
                        *    LDD MAY USE AC14 TO CK TOS IF INDR (CLIB ALREADY EXECUTED, F3=1).
                        *    EXIT WITH SP0_E, OPND_(E), F1 IF F IN TOS (SP1_SM+SR-E), F2=INCM,
                        *      F3 IF INDR.
                        *    BUSOPS USE DB-BNK TO SPECIFY DB OR S-BNK FOR DB OR Q,S REL ADDRS.
                        *
0042    37777777777     AC1S               ADD                       S REL ENTRY
0043    16137777561     AC1D SR   UBUS ADD        BSP0 ROD            DB,Q REL ENTRY
0044    23767117776     AC12 UBUS SM   CAD                  NCRY      E>SM?
```

```
***  WARNING (16)  ***  BOUNDS TEST WITH RRZ,RLZ,LRZ,LLZ DOES A CAD
  165           0045  16306504141          SR    UBUS BNDT RRZ  SP1  CTF  CRRY       YES; BNDV IF NPRV AND
  166                                *                                   SR<E-SM, (SP1-SM+SR-E) SR>=E-SM?
  167           0046  34766717775          SP0   DL   BNDT                 JLUI       NO; E>=DL? DONE IF NOT INDR
  168           0047  37766140055                    JMP  AC14              F1         ELSE JMP IF SM<E<=SM+SR
  169           0050  22777777347    AC1A  XC   DB   ADD             CLIB
  170           0051  26137777576          UBUS OPND ADD        RSP0 ROD              INDR; E = (E) + XC + DB
  171           0052  23767117076          UBUS SM   CAD             SF3  NCRY       SF3; E>SM?
***  WARNING (16)  ***  BOUNDS TEST WITH RRZ,RLZ,LRZ,LLZ DOES A CAD
  172           0053  16306504141          SR    UBUS BNDT RRZ  SP1  CTF  CRRY       YES; BNDV IF NPRV AND
  173                                *                                   SR<E-SM, (SP1-SM+SR-E) SR>=E-SM?
  174           0054  34766717455          SP0   DL   BNDT             CF1  JLUI       NO; CF1, CK E>=DL IF NPRV
  175           0055  37762350272    AC14            JSB  TSCK              UNC        SM<E<=SM+SR; CHECK TOS
  176           0056  37777717777                    ADD                    JLUI       DONE IF CLIB OR NOT INDR
  177           0057  37766350050                    JMP  AC13              UNC
  178                                *
  179                                *
  180                                *    AC1P IS THE P REL ADDR COMPUTATION ROUTINE
  181                                *      FOR ADDM,CMPM,LDX,LOAD,MPYM,SUBM.
  182                                *    ENTERED WITH SBUS=P+- RBUS=PADD+(XC IF NOT INDR).
  183                                *    EXIT WITH SP1=E, OPND=(E).
  184                                *
  185           0060  37777777777    AC1B            ADD                            E=E-1 SINCE P
  186           0061  37107377376          UBUS      CAD       BSP1 ROP               POINTS TO NIR, READ (E)
  187           0062  16766777760          PL    UBUS BNDT                          CHECK PL>=E, E>=PB IF NPRV;
  188           0063  36766717774          SP1   PB   BNDT                 JLUI       DONE IF NOT INDR
  189           0064  01777777347          XC    SP1  ADD             CLIB          E=(E)+E+XC
  190           0065  26117777376          UBUS OPND ADD        BSP1 ROP              READ (E)
  191           0066  16766777760          PL    UBUS BNDT                          CHECK PL>=E IF NPRV
  192           0067  36766717774          SP1   PB   BNDT                 JLUI       CHECK E>=PB IF NPRV; DONE
  193                                *
  194                                *    AINC IS THE ADDR COMPUTATION ROUTINE (EVENTUALLY USING AC1S,D)
  195                                *    FOR INCM.  DECM USES AC1S,D DIRECTLY; BUT INCM'S OP CODE CAUSES
  196                                *    THE LUT TO ASSUME A P REL ADDR, THOUGH E IS SUPPLIED CORRECTLY.
  197                                *
  198           0070  37317777417    AINC            ADD       SP1  SF2             SAVE ADDR, SF2=INCM
  199           0071  00771047077          CIR   ROMI            7077 NSME       CIR(7:9) = 111 ?
  200           0072  01117767561          SR    SP1  ADD       BSP1 ROD  UNC       YES; S REL, ADD SR, READ E
  201           0073  01177777577                SP1  ADD       BUS  ROD            NO; DB OR Q REL, READ E
  202           0074  01326350044                SP1  JMP  AC12 SP0       UNC       FINISH ADDR COMPUTATION
  203                                *
  204                                *
  205                                *         ADDM, SUBM
  206                                *    DB,Q,S OR P REL ADDR; ENTER WITH OPND=(E)
  207                                *
  208           0075  26302201732    ADDM      OPND JSB  PUL1 SP1       SRZ        FILL A TOS REG IF NEC
  209           0076  00777537777                CIR  ADD                NEG
  210           0077  01675757773          RA    SP1  ADDO      RA        NEXT       (S)=(S)+(E); DONE
  211           0100  01665757773          RA    SP1  SUBO      RA        NEXT       (S)=(S)-(E); DONE
  212                                *
  213                                *
  214                                *         LOAD
  215                                *    DB,Q,S OR P REL ADDR; ENTER WITH OPND=(E)
  216                                *
  217           0101  37762221737    LOAD            JSB  PSHM              SR4        EMPTY A TOS REG IF NEC
```

```
     ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:05 PM

218            0102  26217757757                OPND ADD       PUSH CCA  NEXT      TOS_(E); DONE
219                                     *
220                                     *
221                                     *    AC2S,D IS THE S AND DB,Q REL ADDR COMPUTATION ROUTINE FOR LRA.
222                                     *    IF ENTERED AT AC2S SBUS=SM- RBUS=PADD+(XC IF NOT INDR).
223                                     *    IF ENTERED AT AC2D SBUS=DB+ OR Q+- RBUS=PADD+(XC IF NOT INDR)
224                                     *      INSTEAD OF SR+UBUS.
225                                     *    EXIT WITH RANK1 JLUI WITH SP2_E-DB OR IF INDR WITH SP2_(E)+XC.
226                                     *    BUSOPS USE DB-BNK TO SPECIFY DB OR S-BNK FOR DB OR Q,S REL ADDRS.
227                                     *
228            0103  37777777777      AC2S                ADD                                 S REL ENTRY
229            0104  16337777761      AC2D SR    UBUS ADD        SP0                          DB,Q REL ENTRY
230            0105  22727717776           UBUS DB   SUB        SP2       JLUI               SP2_E-DB, DONE IF NOT INDR
231            0106  37177777575           SP0       ADD        BUS  ROD                     INDR; READ (E)
232            0107  23767117355           SP0  SM   CAD                  CLIB NCRY          CLIB; E>SM?
***  WARNING (16)  ***  BOUNDS TEST WITH SRZ,RLZ,LRZ,LLZ DOES A CAD
233            0110  16306714141           SR   UBUS BNDT RRZ  SP1  CTF                       YES; BNDV IF NPRV AND
234                                     *                                  SR<E-SM,  (SP1_SM+SR-E) SR>=E-SM?
235            0111  37762140272                JSB  TSCK                   F1                CHECK TOS IF SM<E<=SM+SR
236            0112  34766777775           SP0  DL   BNDT                                     CHECK E>=DL IF NPRV
237            0113  26737717767           XC   OPND ADD        SP2       JLUI               SP2_XC+(E); DONE
238                                     *
239                                     *    AC2P IS THE P REL ADDR COMPUTATION ROUTINE FOR LRA.
240                                     *    ENTERED WITH SBUS=P+- RBUS=PADD+(XC IF NOT INDR).
241                                     *    EXIT WITH RANK1 JLUI WITH SP2_E-PB OR IF INDR WITH SP2_SP2+XC+(E).
242                                     *
243            0114  37777777777      AC2P                ADD                                 NOTE THAT P POINTS TO NIR
244            0115  36727317776           UBUS PB   CAD        SP2       JLUI               SP2_E-1-PB;DONE IF NOT INDR
245            0116  36137777376           UBUS PB   ADD        BSP0 ROP                     INDR; E_E-1, READ (E)
246            0117  16766777340           PL   UBUS BNDT                 CLIB               CHECK PL>=E IF NPRV
247            0120  36766777775           SP0  PB   BNDT                                     CHECK E>=PB IF NPRV
248            0121  16777777767           XC   UBUS ADD
249            0122  26737717776           UBUS OPND ADD        SP2       JLUI               SP2_XC+E-PB+(E)
250                                     *
251                                     *         LRA
252                                     *    DB,Q,S OR P REL ADDR; ENTER WITH SP2=REL ADDR TO BE LOADED
253                                     *
254            0123  37762221737      LRA                 JSB  PSHM                  SR4       EMPTY A TOS REG IF NEC
255            0124  35217757777                SP2  ADD        PUSH      NEXT               TOS_REL ADDR, NEXT
256                                     *
257                                     *
258                                     *    AC3S,D IS THE ADDR COMPUTATION ROUTINE FOR LDD.
259                                     *    IF ENTERED AT AC3S SBUS=SM- RBUS=PADD+(XC*2 IF NOT INDR).
260                                     *    IF ENTERED AT AC3D SBUS=DB+ OR Q+- RBUS=PADD+(XC*2 IF NOT INDR)
261                                     *      INSTEAD OF SR+UBUS.
262                                     *    EXIT WITH SP0_E, OPND_(E), F1 IF E IN TOS(SP1_SM+SR-E), F3 IF INDR.
263                                     *    BUSOPS USE DB-BNK TO SPECIFY DB OR S-BNK FOR DB OR Q,S REL ADDRS.
264                                     *
265            0125  37777777777      AC3S                ADD                                 S REL ENTRY
266            0126  16137717561      AC3D SR    UBUS ADD        BSP0 ROD                     DB,Q REL ENTRY
267            0127  23767117776           UBUS SM   CAD                  NCRY               E>SM?
***  WARNING (16)  ***  BOUNDS TEST WITH SRZ,RLZ,LRZ,LLZ DOES A CAD
268            0130  16306504141           SR   UBUS BNDT RRZ  SP1  CTF  CRRY               YES; BNDV IF NPRV AND
269                                     *                                  SR<E-SM,  (SP1_SM+SR-E) SR>=E-SM?
270            0131  34766717775           SP0  DL   BNDT                 JLUI               NO; E>=DL? DONE IF NOT INDR
```

```
  271          0132  37762140272                    JSB  TSCK          F1       ELSE CK TOS IF SM<E<=SM+SR
  272          0133  37777712767          XC        ADD  SL1           JLUI     DONE IF NOT INDR
  273          0134  22777777356          UBUS DB   ADD              CLIB       INDR; E_(E)+XC*2+DB
  274          0135  26137777576          UBUS OPND ADD       BSP0 ROD          READ (E)
  275          0136  23767117076          UBUS SM   CAD             SF3  NCRY   SF3; E>SM?
*** WARNING (16) *** BOUNDS TEST WITH RRZ,RLZ,LRZ,LLZ DOES A CAD
  276          0137  16306504141          SR   UBUS BNDT RRZ  SP1   CTF  CRRY   YES; BNDV IF NPRV AND
  277                                  *                                SR<E-SM, (SP1_SM+SR-E) SR>=E-SM?
  278          0140  34766717455          SP0  DL   BNDT             CF1  JLUI   NO; CF1, CK E>=DL IF NPRV
  279          0141  37766300055                    JMP  AC14          UNC      SM<E<=SM+SR; CHECK TOS
  280                                  *
  281                                  *         LOAD DOUBLE WORD
  282                                  *    DB,Q OR S REL ADDR; ENTER WITH SP0=E, OPND=(E), F3 IF INDR.
  283                                  *    LDPP,LDPN,LDEA EXIT THROUGH LDD2 TO PUSH DOUBLE WORD AND SET CCA.
  284                                  *
  285          0142  23767517455     LDD  SP0  SM   SUB             CF1  NCRY   CF1; E+1>SM?
*** WARNING (16) *** BOUNDS TEST WITH RRZ,RLZ,LRZ,LLZ DOES A CAD
  286          0143  16306704141          SR   UBUS BNDT RRZ  SP1   CTF        YES; BNDV IF NPRV AND
  287                                  *                                SR<E+1-SM, (SP1_SM+SR-E+1) SR>=E+1-SM?
  288          0144  26722221737          OPND JSB  PSHM SP2             SR4    EMPTY A TOS REG IF NEC
  289          0145  37176677575          SP0  INC       BUS  ROD  SRL3  E_E+1; READ SECOND WORD
  290          0146  37762361737                 JSB  PSHM                UNC    EMPTY A TOS REG IF NEC
  291          0147  37762140272                 JSB  TSCK                F1     CHECK TOS IF SM<E<=SM+SR
  292          0150  35217407757     LDD2 SP2  ADD       PUSH CCA  ZERO   PUSH FIRST WORD, CCA
  293          0151  26217757777          OPND ADD       PUSH      NEXT   IF FIRST WORD=ZERO,
  294          0152  26217757657          OPND ADD       PUSH CCZ  NEXT     CCZ ON SECOND WORD
  295                                  *
  296                                  *
  297                                  *    AC4S,D IS THE ADDR COMPUTATION ROUTINE FOR STOR.
  298                                  *    IF ENTERED AT AC4S SBUS=SM- RBUS=PADD+(XC IF NOT INDR).
  299                                  *    IF ENTERED AT AC4D SBUS=DB+ OR Q+- RBUS=PADD+(XC IF NOT INDR)
  300                                  *      INSTEAD OF SR+UBUS.
  301                                  *    EXIT WITH SP0_E, F1_UNDF, F3 IF INDR.
  302                                  *    BUSOPS USE DB-BNK TO SPECIFY DB OR S-BNK FOR DB OR Q,S REL ADDRS.
  303                                  *
  304          0153  37777777777     AC4S           ADD                      S REL ENTRY
  305          0154  16337777761     AC4D SR   UBUS ADD       SP0            DB,Q REL ENTRY
  306          0155  34766717776          UBUS DL   BNDT                JLUI   E>=DL? DONE IF NOT INDR
  307          0156  37177777575          SP0       ADD       BUS  ROD       READ (E)
  308          0157  23767117355          SP0  SM   CAD             CLIB NCRY  CLIB; E>SM?
*** WARNING (16) *** BOUNDS TEST WITH RRZ,RLZ,LRZ,LLZ DOES A CAD
  309          0160  16306704141          SR   UBUS BNDT RRZ  SP1   CTF        YES; BNDV IF NPRV AND
  310                                  *                                SR<E-SM, (SP1_SM+SR-E) SR>=E-SM?
  311          0161  37762140272                    JSB  TSCK          F1       CHECK TOS IF SM<E<=SM+SR
  312          0162  22777777767          XC   DB   ADD
  313          0163  26337777776          UBUS OPND ADD       SP0            E = (E) + XC + DB
  314          0164  34766717076          UBUS DL   BNDT             SF3  JLUI   SF3, CK E>=DL IF NPRV
  315                                  *
  316                                  *
  317                                  *    AC5S,D IS THE ADDR COMPUTATION ROUTINE FOR STD.
  318                                  *    IF ENTERED AT AC5S SBUS=SM- RBUS=PADD+(XC*2 IF NOT INDR).
  319                                  *    IF ENTERED AT AC5D SBUS=DB+ OR Q+- RBUS=PADD+(XC*2 IF NOT INDR)
  320                                  *      INSTEAD OF SR+UBUS.
  321                                  *    EXIT WITH SP0_E, F1_UNDF, F3 IF INDR.
  322                                  *    BUSOPS USE DB-BNK TO SPECIFY DB OR S-BNK FOR DB OR Q,S REL ADDRS.
```

```
  ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:05 PM

                         *
          0165  37777777777  ACS0           ADD                                  S REL ENTRY
          0166  16337777761  ACS1  SR   UBUS ADD        SP0                      DB,Q REL ENTRY
          0167  34766717776        UBUS DL   BNDT                    JLUI       E>=DL? DONE IF NOT INDR
          0170  37177777575        SP0       ADD        BUS  ROD                 READ (E)
          0171  23767117355        SP0  SM   CAD                    CLIB NCRY  CLIB; E>SM?
***  WARNING (16)  ***  BOUNDS TEST WITH RRZ,RLZ,LRZ,LLZ DOES A CAD
          0172  16306774141        SR   UBUS BNDT RRZ  SP1  CTF                 YES; BNDV IF NPRV AND
                         *                                        SR<E-SM, (SP1_SM+SR-E) SR>=E-SM?
          0173  37337592767        XC        ADD  SL1  SP0       NF1            SP0_XC*2
          0174  37762360272                  JSB  TSCK           UNC            CHECK TOS IF SM<E<=SM+SR
          0175  22777777775        SP0  DB   ADD
          0176  26337777776        UBUS OPND ADD        SP0                      E_(E)+XC*2+DB
          0177  34766717076        UBUS DL   BNDT                 SF3  JLUI     SF3, CK E>=DL IF NPRV
                         *
                         *          STORE DOUBLE WORD
                         *     DB,Q OR S REL ADDR; ENTER WITH SP0=E, F3 IF INDR
                         *
          0200  37762291732  STD             JSB  PUL1           SRLZ           FILL A TOS REG IF NEC
          0201  23767517455        SP0  SM   SUB                  CF1  NCRY     CF1; E+1>SM?
***  WARNING (16)  ***  BOUNDS TEST WITH RRZ,RLZ,LRZ,LLZ DOES A CAD
          0202  16306774141        SR   UBUS BNDT RRZ  SP1  CTF                 YES; BNDV IF NPRV AND
                         *                                        SR<E+1-SM, (SP1_SM+SR-E-1) SR>=E+1-SM?
          0203  37176777555        SP0       INC        BUS  WRD                 STORE (S) IN
          0204  33177547457             RA   ADD        BUS  DPOP F1               MEM AT E+1, S_S-1
          0205  37766360211                  JMP  STOR           UNC            JMP IF E+1 NOT IN TOS REGS
          0206  37762360272                  JSB  TSCK           UNC            CHECK FOR E+1 IN TOS REGS
          0207  37307157774        SP1       CAD        SP1      NF1            ADJUST SP1 FOR POP
          0210  30077777777             RD   ADD        MREG                    STORE IF E+1 IN TOS REGS
                         *
                         *          STOR
                         *     DB,Q OR S REL ADDR; ENTER WITH SP0=E, F3 IF INDR,
                         *     PSTA AND STB EXIT THROUGH STR2 IF STORING IN TOS,
                         *
          0211  37722201732  STOR            JSB  PUL1 SP2       SRZ            SP2_0; FILL A TOS IF NEC
          0212  23767117455        SP0  SM   CAD                  CF1  NCRY     CF1; E>SM?
***  WARNING (16)  ***  BOUNDS TEST WITH RRZ,RLZ,LRZ,LLZ DOES A CAD
          0213  16306774141        SR   UBUS BNDT RRZ  SP1  CTF                 YES; BNDV IF NPRV AND
                         *                                        SR<E-SM, (SP1_SM+SR-E) SR>=E-SM?
          0214  37177547555        SP0       ADD        BUS  WRD  F1             STORE IN MEM
          0215  33177757457             RA   ADD        BUS  DPOP NEXT           S_S-1, DONE IF NOT IN TOS
          0216  33177777437             RA   ADD        BUS  DATA                E COULD BE IN TOS REGS
          0217  37762360272                  JSB  TSCK           UNC            CHECK FOR E IN TOS REGS
          0220  37766150645                  JMP  DEL            NF1            POP STK IF NOT IN TOS REGS
          0221  35077757573  STR2 RA   SP2  ADD        MREG POP  NEXT           ELSE STORE IN TOS, POP STK
                         *
                         *
                         *     ALSB IS THE ADDR COMPUTATION ROUTINE FOR LDB AND STB,
                         *     ENTERED WITH SBUS=DB+, SM- OR Q+- RBUS=PADD+(XC/2 IF NOT INDR),
                         *     EXIT WITH SP0=E, OPND_(E), F2 IF RH BYTE,
                         *       F1 SET AND SP1_SM+SR-E IF E IS IN THE TOS REGS,
                         *     BUSOPS USE DB-BNK TO SPECIFY DB OR S-BNK FOR DB OR Q,S REL ADDRS,
                         *
          0222  37337777777  ALSB            ADD        SP0                      SP0_E ASSUMING DB,Q
          0223  16317777761        SR   UBUS ADD        SP1                      SP1_E IF S REL
```

```
      ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:05 PM

375           0224  00771047077                CIR  ROMI                7077 NSME
376           0225  01326350226                SP1  JMP  *+1  SP0          UNC      SP0=E IF S REL (2C JMP)
377           0226  23307107155          ALS1  SP0  SM   CAD       SP1  CTF  CRRY     SP1=E-SM-1; SF1 IF E>SM
378           0227  34767507775                SP0  DL   SUB                 CRRY     E>=DL?
379           0230  37762350256                     JSB  ALS3                UNC      JSB IF E>SM OR E<DL
380           0231  14777547367                XC   CTRL ADD                LBF  F1  SF2 IF RH BYTE;
381           0232  37177777575                SP0       ADD       BUS  ROD          READ E IF NOT IN TOS
382           0233  37766240253                     JMP  ALS2                INDR     JMP IF INDR
383           0234  00761620000                     CIR  ROMN                020000
384           0235  16766010242                     UBUS JMP  STB                NZRO     JMP IF STB
385                                       *
386                                       *          LDB
387                                       *     ENTER WITH OPND=(E), F2 IF RH BYTE
388                                       *
389           0236  37762221737          LDB              JSB  PSHM                SR4      EMPTY A TOS REG IF NEC
390           0237  37777567777                           ADD                      F2
391           0240  26217750017                     OPND ADD  LRZ  PUSH CCB  NEXT     TOS_LH BYTE, CCB, DONE
392           0241  26217754017                     OPND ADD  RRZ  PUSH CCB  NEXT     TOS_RH BYTE, CCB, DONE
393                                       *
394                                       *          STB
395                                       *     ENTER WITH SP0=E, OPND=(E), F2 IF RH BYTE,
396                                       *     F1 SET AND SP1=SM+SR-E IF E IS IN TOS REGS.
397                                       *
398           0242  26737501777          STB      OPND ADD  LLZ  SP2       F2       SAVE ORIGINAL LH BYTE IF F2
399           0243  26737774777                OPND ADD  RRZ  SP2                ELSE SAVE ORIGINAL RH BYTE
400           0244  33762201732                RA   JSB  PUL1                SRZ      FILL A TOS REG IF NEC;
401                                       *                                           STK WILL BE POPPED SO E=S
402                                       *                                           AND SR=0 DOES NOT MATTER.
403           0245  16677504777                UBUS ADD  RRZ  RA        F2       STORE IN RH OF E IF F2
404           0246  33677775777                RA   ADD  RLZ  RA                 ELSE STORE IN LH OF E
405           0247  37766140221                     JMP  STR2                F1       STORE IN TOS IF F1
406           0250  37177777555                SP0       ADD       BUS  WRD          STORE IN MEM
407           0251  35177757453                RA   SP2  ADD       BUS  DPOP NEXT    SEND DATA, DONE
408           0252  37777777777                           ADD
409                                       *
410           0253  26777773067          ALS2 XC   OPND ADD  SR1       SF3           INDR; SF3
411           0254  22337777356                UBUS DB   ADD       SP0  CLIB         E=[(E)+XC]/2+DB, CLIB
412           0255  26346350226                     OPND JMP  ALS1 CTRL       UNC      CTR(5)_(ORG E)(15), CHECK E
413                                       *
414           0256  01307117761          ALS3 SR   SP1  CAD       SP1       NCRY     SP1_SM+SR-E; SR>=E-SM?
415           0257  37766140272                     JMP  TSCK                F1       YES, CK TOS IF E>SM
416           0260  03777777617                RBR  ADD                 S             SPLIT STACK IF S-BANK IS
417           0261  03767407416                UBUS RBR  SUB                 DB   ZERO      NOT THE SAME AS DB-BANK
418           0262  22777707457                     DB   ADD                 CF1  RSB      IF SPLIT STK CF1, RETURN
419           0263  34767517456                UBUS DL   SUB                 CF1  NCRY     CF1; DB>=DL?
420           0264  22767507762                Z    DB   SUB                      CRRY     Z>=DB?
421           0265  37777707775                SP0       ADD                      RSB      NO; SPLIT STK, RETURN
422           0266  16331700000                     UBUS ROM       SP0  100000          E=E+32K
423           0267  23767117776                UBUS SM   CAD                      NCRY     E>SM?
***  WARNING (16)  ***  BOUNDS TEST WITH RZ,RLZ,LRZ,LLZ DOES A CAD
424           0270  16306504141                SR   UBUS BNDT RRZ  SP1  CTF  CRRY     YES; BNDV IF NPRV AND
425                                       *                                SR<E-SM, (SP1_SM+SR-E) SR>=E-SM?
426           0271  34766707775                SP0  DL   BNDT                RSB      NO; CK E>=DL IF NPRV, RET
427                                       *
428                                       *     SM<E<=SM+SR; TEST FOR E IN TOS REGS.
```

```
  ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2105 PM

                          *     ENTER AT TSCK TO CK ADDR MODE (IN CIR) AND IF DB REL OR F3 COMPARE
                          *       DB-BNK WITH S-BNK.  ENTER AT TSC1 TO COMPARE RBUS WITH S-BNK.
                          *     OPND_MREG IF E IN TOS REGS, ELSE CF1;
                          *       IF ENTERED AT TSCK RETURN MAY BE A RANK1 RSB WITH OPND_MREG.
                          *
*** WARNING ( 2)  *** RBR CONFLICTS WITH PREFETCH ON INSTR ENTRY
0272  03777747417       TSCK      RBR  ADD             DB   F3
0273  00773063776            UBUS CIR  IOR  SR1             BITR       NF3 AND Q OR S REL,
*** WARNING ( 2)  *** RBR CONFLICTS WITH PREFETCH ON INSTR ENTRY
0274  03767417605       TSC1 RBUS RBR  SUB             S    NZRO         OR DB-BNK=S-BNK?
0275  37177707623            MREG      ADD       BUS  OPND RSB        YES, OPND_(E) FROM TOS REGS
0276  37777707457                      ADD             CF1  RSB        ELSE CF1, (E) IN MEM
                          *
                          *
                          *     LDPP, LDPN
                          *   LOAD DOUBLE WORD AT P+N OR P-N; PADD = +-N
                          *
0277  20771777777       LDPA      P    ROM             177777          P POINTS TO NIR
0300  16117777364            PADD UBUS ADD       RSP1 ROP             E = P - 1 +- PADD
0301  36766637776            UBUS PB   BNDT                SRN4       CHECK E>=PB IF NPRV
0302  37762361737                      JSB  PSHM            UNC        EMPTY A TOS REG IF NEC
*** WARNING (16)  *** BOUNDS TEST WITH RZ,RLZ,LRZ,LLZ DOES A CAD
0303  01766674760            PL   SP1  BNDT RR7             SRLA       CHECK PL>E IF NPRV
0304  37762361737                      JSB  PSHM            UNC        EMPTY A TOS REG IF NEC
0305  01116777377                 SP1  INC       RSP1 ROP              READ (E+1)
0306  26726230150                 OPND JMP  LD02 SP2         SRN4       PUSH WDS, SET CC (2C JMP)
                          *
                          *
                          *     PLDA/PSTA
                          *   PADD=CIR(8:15); CHECK MADE FOR E IN TOS REGS
                          *
0307  37726263117       PLSA           JMP  TRP6 SP2         NPRV       SP2_0; PLDA/PSTA ARE PRV
0310  37177777166            X         ADD       BUS  ROA             READ (X)
0311  23767117766            X    SM   CAD                  NCRY       X>SM? YES; SF1 IF SR>=X-SM,
0312  16307377141            SR   UBUS CAD       SP1  CTF                          SP1_SM+SR-X
0313  37762140274                      JSB  TSC1            F1         CK S-BNK=0 IF SM<X<=SM+SR
0314  02766030317                 PADD JMP  PSTA            ODD        JMP IF STORE
0315  37762221737                      JSB  PSHM            SR4        EMPTY A TOS REG IF NEC
0316  26217757757                 OPND ADD       PUSH CCA  NEXT       TOS_(X), CCA, DONE
                          *
0317  37766140221       PSTA           JMP  STR2            F1         STORE IN TOS IF F1
0320  37762201732                      JSB  PUL1            SRZ        FILL A TOS REG IF NEC
0321  37177777146            X         ADD       BUS  WRA             STORE IN MEM,
0322  33177757457                 RA   ADD       BUS  DPOP NEXT         S_S-1, DONE
                          *
                          *
                          *     LSEA/LDEA/SSEA/SDEA
                          *   LOAD SINGLE OR DOUBLE WORD FROM ABS ADDR.
                          *   STORE SINGLE OR DOUBLE WORD AT ABS ADDR; DELETE DATA.
                          *   ENTER WITH SR>=2; E(S) CANNOT POINT TO THE INSTR PARAMS ON THE TOS.
                          *
0323  37766263117       LSAA           JMP  TRP6            NPRV       LSEA,LDEA,SSEA,SDEA ARE PRV
0324  00775123377                 CIR  CRS  SR1       LBF  POS        SF2 IF LDEA OR SDEA
0325  37766360335                      JMP  LSA5            UNC        JMP IF SSEA OR SDEA
0326  37762221737                      JSB  PSHM            SR4        EMPTY A TOS REG IF NEC
```

```
      ADDRESS   CONTENTS      LABEL  RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:05 PM

481     0327  32157677017             RB   ADD       SBR  ABS  SRL3      ABS-BANK_(S-1)
482     0330  37762361737                  JSB  PSHM                UNC      EXACTLY 2 TOS REGS FILLED
483     0331  33177577177             RA   ADD       BUS  ROA  NF2       READ ((S-1),(S))
484     0332  33176767177             RA   INC       BUS  ROA  UNC       LDEA; READ ((S-1),(S)+1)
485     0333  26217757757             OPND ADD       PUSH CCA  NEXT      LSEA; TOS_DATA, CCA, DONE
486     0334  16726210150             UBUS JMP  LDD2 SP2       SRNZ      LDEA; PUSH, SET CC (2C JMP)
487                             *
488     0335  31762271732     LSA5    RC   JSB  PUL1                SRL3      FILL A TOS REG IF NEC
489     0336  16157567017             UBUS ADD       SBR  ABS  F2        ABS-BANK_(S-2)
490     0337  37766360344                  JMP  LSA6                UNC        AND JMP IF SSEA
491     0340  30762231732             RD   JSB  PUL1                SRN4      ELSE FILL 4 TOS REGS
492     0341  16157777017             UBUS ADD       SBR  ABS                  AND ABS-BANK_(S-3)
493     0342  31176777157             RC   INC       BUS  WRA                  (ABS-BNK,(S-2)+1)_(S)
494     0343  33177767457             RA   ADD       BUS  DPOP UNC             S_S-1
495     0344  37762221737     LSA6         JSB  PSHM                SR4       EXACTLY 3 TOS REGS FILLED
496     0345  32177777157             RB   ADD       BUS  WRA                  (ABS-BNK,(S-1))_(S),
497     0346  33177757457             RA   ADD       BUS  DPOP NEXT            S_S-1, DONE
498                             *
499                             *
500                             *        LST
501                             *    LOAD FROM SYSTEM TABLES
502                             *    IF K=0 THEN (S)_(X+1000+((S)+1000)) ELSE TOS_(X+1000+(K+1000))
503                             *    ENTER WITH SR>=1, PADD=K=CIR(12:15); NO CHECK FOR E IN TOS REGS
504                             *
505     0347  37731601000     LST          ROM       SP2  001000          SP2_1000
506     0350  02777417777             PADD ADD                   NZRO
507     0351  35177767173         RA   SP2  ADD       BUS  ROA  UNC       IF K=0 READ ((S)+1000)
508     0352  35177767164         PADD SP2  ADD       BUS  ROA  UNC       ELSE READ (K+1000)
509     0353  35777777566         X    SP2  ADD            POP           UBUS_X+1000; IF K=0 S_S-1
510     0354  16762263117             UBUS JSB  TRP6                NPRV      LST IS PRV
511     0355  26177637176         UBUS OPND ADD       BUS  ROA  SRN4      READ (X+1000+OPND)
512     0356  37762361737                  JSB  PSHM                UNC       EMPTY A TOS REG IF NEC
513     0357  26217757757             OPND ADD       PUSH CCA  NEXT      PUSH DATA, CCA, DONE
514                             *
515                             *
516                             *        SST
517                             *    STORE INTO SYSTEM TABLES
518                             *    IF K=0 THEN (X+1000+((S)+1000))_(S-1), S_S-2
519                             *    ELSE (X+1000+(K+1000))_(S), S_S-1
520                             *    ENTER WITH SR>=2, PADD=K=CIR(12:15); NO CHECK FOR E IN TOS REGS
521                             *
522     0360  37731601000     SST          ROM       SP2  001000          SP2_1000
523     0361  02777417777             PADD ADD                   NZRO
524     0362  35177767173         RA   SP2  ADD       BUS  ROA  UNC       IF K=0 READ ((S)+1000)
525     0363  35177767164         PADD SP2  ADD       BUS  ROA  UNC       ELSE READ (K+1000)
526     0364  35777777566         X    SP2  ADD            POP           UBUS_X+1000; IF K=0 S_S-1
527     0365  16766263117             UBUS JMP  TRP6                NPRV      SST IS PRV
528     0366  26177777156         UBUS OPND ADD       BUS  WRA            WRITE AT (X+1000+OPND)
529     0367  33177757457             RA   ADD       BUS  DPOP NEXT      WRITE (S), S_S-1, DONE
```

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:05 PM

                        &              SECTOR 1
                        *
                        *          BRANCH INSTRS: BR/BCC, BOV , BNOV, BCY , BNCY, IXBZ,
                        *                         DXBZ  , IABZ, DABZ, BRE , BRO , CPRB
                        *
                        *          LOOP CONTROL BRANCH INSTRS: TBA/MTBA/TBX/MTBX
                        *
                        *          IMMEDIATE INSTRS: CMPI, CMPN, DIVI
                        *
                        *          STACKOPS: INCX, DECX, INCA, DECA, INCB, DECB;
                        *                    DTST, NOP ;
                        *                    XAX , XBX , XCH , DXCH, CAB;
                        *                    LDXA, ADXA, STAX, ADAX,
                        *                    LDXB, ADXB, STBX, ADBX;
                        *                    ADD , SUB , NEG , CMP ;
                        *                    DADD, DSUB, DNEG, DCMP, ZROB;
                        *                    LCMP, DDEL, DEL , DELB,
                        *                    LADD, LSUB, NOT , LMPY, LDIV;
                        *                    MPY , MPYL, DIV , DIVL
                        *                    (OR,XOR,AND ARE IN SECTOR 0;
                        *                     FP STACKOPS,TEST,BTST ARE IN SECTOR 2)
                        *
                        *          MEM REF INSTR: MPYM
                        *
                        *          IMMEDIATE INSTRS: MPYI, LDI , LDNI, LDXI, LDXN, ADXI,
                        *                            SBXI, ORI , XORI, ANDI, ADDI, SUBI
                        *
                        *          STACKOPS: DUP , DDUP;
                        *                    DZRO, ZERO, ZROX
                        *
                        *
                        *          BR/BCC
                        *     BR S REL ENTERS AT BRS WITH SBUS=P+- RBUS=PADD.
                        *     BR DB OR Q REL ENTER AT BRD WITH SBUS=DB+ OR Q+- RBUS=PADD.
                        *     BR P REL ENTERS AT BRP WITH SBUS=P+- RBUS=PADD+(XC IF NOT INDR).
                        *     BUSOPS USE DB-BNK TO SPECIFY DB OR S-BNK FOR DB OR Q,S REL ADDRS.
                        *     BCC ENTERS AT BRS OR BRD WITH PADD= +-CIR(11:15) AND EXITS TO BCC1.
                        *     ALL CONDITIONAL BRANCHES USE BCC2 TO BRANCH; BR DB,Q,S USE BCC4.
                        *
                        &0400
0400  37777777777       BRS             ADD                                 S REL ENTRY
0401  16337647761       BRD  SR   UBUS  ADD       SP0       INDR            DB,Q REL ENTRY
0402  00766300422              CIR  JMP  BCC1              UNC             JMP IF BCC INSTR
0403  37177777575         SP0        ADD       PUS  ROD                    READ INDR ADDR = (E)
0404  23767117775         SP0  SM    CAD                   NCRY            E>SM?
***  WARNING (16)  ***  BOUNDS TEST WITH RZ,RLZ,LRZ,LLZ DOES A CAD
0405  16306504141         SR   UBUS  BNDT  RRZ  SP1  CTF  CRRY            YES; BNDV IF NPRV AND
                        *                                 SR<E-SM, (SP1-SM+SR-E) SR>=E-SM?
0406  34766767775         SP0  DL    BNDT                  UNC             NO; CHECK E>=DL IF NPRV
0407  37762300272              JSB  TSCK                  UNC             SM<E<=SM+SR; CHECK TOS
0410  36317777767         XC   PB    ADD       SP1                         SEND (E=(E)+XC+PB)
0411  37766300432              JMP  BCC4                  UNC               TO NIR, CHECK BOUNDS
                        *
0412  37777777777       BRP             ADD                                 NOTE P POINTS TO NIR
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS
583               0413    37107377276         UBUS      CAD       BSP1 RONP          SEND (E) TO OPND AND NIR
584               0414    16764777760   BRP2  PL   UBUS UBNT                        CHECK PL>=E
585               0415    36764647774         SP1  PB   UBNT                  INDR  CHECK E>=PB; INDR?
586               0416    37416757774         SP1       INC       P           NEXT  NO; POINT P TO E+1, DONE
587               0417    01777777347         XC   SP1  ADD             CLIB       YES; CLIB, E_E+(E)+XC
588               0420    26117777316         UBUS OPND ADD       BSP1 RNP          SEND (E) TO NIR
589               0421    16766360414               UBUS JMP  BRP2                UNC   CHECK BOUNDS, SET P
590                                           *
591               0422    27763417776   BCC1  UBUS CC   AND                       NZRO  BCC; CIR(7:9) AND CC = 0 ?
592               0423    37777757777               ADD                           NEXT  YES, NEXT
593                                           *
594               0424    20771777776   BCC2        P    ROM             177776         P POINTS TO NIR+1
595               0425    16117547264         PADD UBUS ADD       BSP1 RONP F1      SEND (E) TO OPND AND NIR
596                                           *     ENTER WITH F1 SET AND CTR NOT 77 TO INHIBIT CIR(4) TEST
597               0426    00371377777               CIR  ROMI       CTRH 173777       CTR_77 IF CIR(4)=1 (INDR)
598               0427    01764777760   BCC3  PL   SP1  UBNT                        CHECK PL>=E
599               0430    36764737774         SP1  PB   UBNT                  CTRM  CHECK E>=PB; INDR?
600               0431    01416757777               SP1  INC       P           NEXT  NO; POINT P TO E+1, DONE
601                                           *
602               0432    26117777314   BCC4  SP1  OPND ADD       BSP1 RNP          YES; E_E+(E), (E) TO NIR
603               0433    37346360427               JMP  BCC3 CTRL                UNC   CTR_0; CK BOUNDS, SET P
604                                           *
605                                           *
606                                           *          BOV, BNOV, BCY, BNCY, IXBZ, DXBZ
607                                           *     PADD = DISPLACEMENT = +-CIR(11:15)
608                                           *
609               0434    24371377777   BOV         STA  ROMI       CTRH 173777       CTR_77 IF STA(4)=1
610               0435    37777777637               ADD                 CLO           CLEAR OVFL
611               0436    37766360424               JMP  BCC2                   CTRM  BRANCH IF OVFL=STA(4)
612               0437    37777757777               ADD                           NEXT    WAS SET, ELSE NEXT
613                                           *
614               0440    24761604000   BNOV        STA  ROMN             004000       MASK OVFL BIT = STA(4)
615               0441    16766000424               UBUS JMP  BCC2                ZERO  BRANCH IF ZERO
616               0442    37777757637               ADD                 CLO   NEXT  ELSE CLO, NEXT
617                                           *
618               0443    24777453537   BNCY        STA  ADD  SR1       CCRY BIT6     CCRY; BRANCH IF CRY=STA(5)
619               0444    37766360424               JMP  BCC2                   UNC     WAS NOT SET ELSE FALL THR
620                                           *                                       BCY, WAITING FOR PREFETCH
621               0445    24767053537   BCY         STA  CAD  SR1       CCRY BIT6     CCRY
622               0446    37766360424               JMP  BCC2                   UNC   BRANCH IF CRY=STA(5)
623               0447    37777757777               ADD                           NEXT    WAS SET, ELSE NEXT
624                                           *
625               0450    37554417766   IXBZ  X           INCO      X           NZRO  X_X+1, CCA,OVFL; ZERO?
626               0451    37766360424               JMP  BCC2                   UNC   YES, BRANCH
627               0452    37777757777               ADD                           NEXT  ELSE NEXT
628                                           *
629               0453    37545017766   DXBZ  X           CADO      X           NZRO  X_X-1, CCA,OVFL; ZERO?
630               0454    37766360424               JMP  BCC2                   UNC   YES, BRANCH
631               0455    37777757777               ADD                           NEXT  ELSE NEXT
632                                           *
633                                           *
634                                           *          IABZ, DABZ, BRE, BRO
635                                           *     ENTER WITH SR>=1, PADD = DISPLACEMENT = +-CIR(11:15)
636                                           *
637               0456    33674417777   IABZ        RA   INCO      RA          NZRO  (S)_(S)+1, CCA,OVFL; ZERO?
```

```
      ADDRESS   CONTENTS      LABEL  RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:05 PM

0457  37766390424                          JMP  BCC2                UNC       YES, BRANCH
0460  37777757777                          ADD                      NEXT      ELSE NEXT
                          *
0461  37665017773  DAB7    RA             CADO       RA            NZRO      (S)_(S)-1, CCA,OVFL; ZERO?
0462  37766390424                          JMP  BCC2                UNC       YES, BRANCH
0463  37777757777                          ADD                      NEXT      ELSE NEXT
                          *
0464  33777437577  BRE         RA    ADD                 POP  ODD       S_S-1
0465  37766390424                    JMP  BCC2                UNC       BRANCH IF (S) WAS EVEN
0466  37777757777                    ADD                      NEXT      ELSE NEXT
                          *
0467  33777427577  BRO         RA    ADD                 POP  EVEN      S_S-1
0470  37766390424                    JMP  BCC2                UNC       BRANCH IF (S) WAS ODD
0471  37777757777                    ADD                      NEXT      ELSE NEXT
                          *
                          *
                          *         CPRB
                          *    BRANCH IF (SIGNED) (S-1)<=X<=(S)
                          *    ENTER WITH SR>=2, PADD = +-CIR(11:15)
                          *
0472  33767077566  CPRB  X     RA    CAD                 POP  NOFL      TEST UPPER BOUND; IF SAME
0473  16767377777               UBUS CAD                              SIGN CCG IF X>UPPER BOUND
0474  16777537577               UBUS ADD                 POP  NEG       ELSE CCG IF X POS
0475  37777757706         X          ADD                 CCG  NEXT      (S_S-2, RB NOW IN RD)
0476  16767077730         RD    UBUS CAD                 CCE  NOFL      TEST LOWER BOUND; IF SAME
0477  16767377777               UBUS CAD                              SIGN CCL IF LOWER BOUND>X
0500  16766130424               UBUS JMP  BCC2                NEG       ELSE CCL IF LOWER BOUND
0501  37777757677                    ADD                 CCL  NEXT      POS, ELSE BRANCH
                          *
                          *
                          *         TBA, MTBA, TBX, MTBX
                          *    TEST AND BRANCH, AND MODIFY,TEST AND BRANCH INSTRS,
                          *    (S)=LIMIT; (S-1)=STEP SIZE; (S-2)=DB REL ADDR OF TEST
                          *      VARIABLE, OR X=TEST VARIABLE.
                          *    ENTRY IS VIA A MEM REF LUT ENTRY; PADD=+-N (CIR(8:15)).
                          *
0502  37762201732  MTB1                   JSB  PUL1                SRZ       FILL 2 TOS REGS IF NEC
0503  37762201732                          JSB  PUL1                SRL2
0504  00761604000               CIR  ROMN                004000
0505  16766000523               UBUS JMP  MTB6                ZERO      JMP IF TBA OR MTBA
0506  32317647766         X     RB   ADD       SP1                INDR      SP1 _ X + STEP SIZE
0507  37317767466         X          ADD       SP1  SF1  UNC      ELSE IF TBX, SP1_X
0510  32557477766         X     RB   ADD       X                  NOFL      MTBX; X _ X + STEP SIZE
0511  37766150520                    JMP  MTB4                     NF1       TERM LOOP IF OVFL AND MTBX
0512  32777777317  MTB2         RB   ADD                 HBF               SF1 IF STEP NEG
0513  01311700000               SP1  ROM       SP1  100000               CONVERT VAR TO LOGICAL
0514  33737547765         RBUS  RA   ADD       SP2       F1               CONVERT LIMIT TO LOGICAL
0515  01767767156         UBUS  SP1  SUB                 CTF  UNC        STEP POS; LIMIT>=VAR?
0516  35767777154         SP1   SP2  SUR                 CTF              STEP NEG; VAR>=LIMIT?
0517  37346140424                    JMP  BCC2 CTRL           F1          YES, BRANCH (CTR_0, F1 SET)
0520  37777577577  MTB4             ADD                      POP  NF2
0521  37777777577                    ADD                      POP       S_S-3 IF TBA OR MTBA (F2)
0522  37777757577                    ADD                      POP  NEXT      S_S-2 IF TBX OR MTBX (NF2)
                          *
0523  37762271732  MTB6             JSB  PUL1                SRL1      TBA, MTBA
```

```
      ADDRESS   CONTENTS      LABEL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:05 PM

693           0524  37762241737                     JSR   PSHM                SR4         EXACTLY 3 TOS REGS FILLED
694           0525  23337777357                 SM  ADD         SP0   CLIB              SP0_SM, CLR MEM REF BNK FF
695           0526  22117777571             RC  DB  ADD         RSP1  ROD               E=RC+DB, READ TEST VAR=(E)
696           0527  16766777775             SP0 UBUS BNDT                               CHECK SM>=E IF NPRV
697           0530  34766777414             SP1 DL  BNDT                SF2             CHECK E>=DL IF NPRV
698           0531  00777493777                 CIR ADD   SR1                   BIT6
699           0532  26306390512                 OPND JMP  MTB2  SP1             UNC         JMP IF TBA
700           0533  01177777557                 SP1 ADD         BUS   WRD                   MTBA; (E)_(E)+RB
701           0534  26117477432             RB  OPND ADD        RSP1  DATA NOFL
702           0535  37766390520                     JMP   MTB4                  UNC         TERMINATE LOOP IF OVFL
703           0536  37766390512                     JMP   MTB2                  UNC         TEST FOR COMPLETION
704                               *
705                               *
706                               *         CMPI, CMPN
707                               *    ENTER WITH SR>=1; PADD=-N FOR CMPI, PADD=N FOR CMPN (CIR(8:15))
708                               *
709           0537  02777477553 CMPI RA   PADD ADD             POPA NOFL      CCA ON (S)-N
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
710           0540  00773357753             RA  CIR   IOR             CCA   NEXT        REVERSE CCL, CCG IF OVFL
711           0541  37777757777                     ADD                 NEXT          (CIR>0 SO NO CCE IF OVFL)
712                               *
713                               *
714                               *         DIVI
715                               *    ENTER WITH SR>=1
716                               *
717           0542  02726003130 DIVI       PADD JMP  TRP4  SP2         ZERO        DIVIDE BY ZERO
718           0543  33317527777             RA  ADD         SP1         POS
719           0544  33307777777             RA  SUB         SP1                     SP1_ABS(U)
720           0545  37772577777                 REPN                    21
721           0546  35764332276         UBUS SP2  DVSB  SL1         INCT CTRM
722           0547  33763137774         SP1 RA  XOR                     NEG         IF SIGN U = SIGN W
723           0550  01677757757             SP1 ADD         RA    CCA   NEXT          THEN (S)_W, CCA, DONE
724           0551  01667757757             SP1 SUB         RA    CCA   NEXT        ELSE (S) _ -W, CCA, DONE
725                               *
726                               *
727                               *         INCX, DECX
728                               *
729           0552  37554757766 INCX X          INCO        X           NEXT        X_X+1; CCA,OVFL
730           0553  37545357766 DECX X          CADO        X           NEXT        X_X-1; CCA,OVFL
731                               *
732                               *
733                               *         INCA, DECA
734                               *    ENTER WITH SR>=1
735                               *
736           0554  33674757777 INCA        RA  INCO        RA          NEXT        (S)_(S)+1; CCA,OVFL
737           0555  37665357773 DECA RA         CADO        RA          NEXT        (S)_(S)-1; CCA,OVFL
738                               *
739                               *
740                               *         INCB, DECB
741                               *    ENTER WITH SR>=2
742                               *
743           0556  32654757777 INCB        RB  INCO        RB          NEXT        (S-1)_(S-1)+1; CCA,OVFL
744           0557  37645357772 DECB RB         CADO        RB          NEXT        (S-1)_(S-1)-1; CCA,OVFL
745                               *
746                               *
```

```
      ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:05 PM

747                           *         DTST, NOP
748                           *     ENTER WITH SR>=2 FOR DTST,
749                           *     VARIOUS INSTRS WITH DOUBLE WORD RESULTS EXIT THROUGH
750                           *       DTST AND DCCA TO SET CC, AND S/C CRRY IF DTST.
751                           *     VARIOUS OTHER INSTRS EXIT THROUGH NOP.
752                           *
753       0560  32777447777   DTST      RB    ADD                 NSME      CLEAR CARRY IF
754       0561  32763127533         RA  RB    XOR           CCRY  POS         HIGH ORDER 17 BITS
755       0562  37777777517             RB    ADD           SCRY            ARE ALL ZEROS OR ALL ONES
756       0563  32777407757   DCCA      RB    ADD           CCA   ZERO      CCA ON (S-1)
757                           *
758       0564  37777757777   NOP             ADD                 NEXT
759                           *                                             IF (S-1)=ZERO,
760       0565  33777757657             RA    ADD           CCZ   NEXT        THEN CCZ ON (S)
761                           *
762                           *
763                           *         XAX, XBX, XCH
764                           *     ENTER WITH SR>=1 FOR XAX, SR>=2 FOR XBX AND XCH
765                           *
766       0566  33557777777   XAX       RA    ADD         X                 X_(S)
767       0567  37677757746         X         ADD         RA    CCA   NEXT  (S)_X, CCA, NEXT
768                           *
769       0570  32557777777   XBX       RB    ADD         X                 X_(S-1)
770       0571  37657757766         X         ADD         RB          NEXT  (S-1)_X, NEXT
771                           *
772       0572  33657777777   XCH       RA    ADD         RB                (S-1)_(S)
773       0573  32677757757             RB    ADD         RA    CCA   NEXT  (S)_(S-1), CCA, NEXT
774                           *
775                           *
776                           *         DXCH
777                           *     ENTER WITH SR=4
778                           *
779       0574  37777777257   DXCH            ADD               INCN        EXCHANGE
780       0575  37777777257                   ADD               INCN          (S-1),(S) AND (S-3),(S-2)
781       0576  37766390563                   JMP   DCCA              UNC     SET CC ON (S-1),(S)
782                           *
783                           *
784                           *         CAB
785                           *     ENTER WITH SR>=3
786                           *     NET RESULT: RD UNCHANGED, RA_RC, RB_RA, RC_RB
787                           *     PERFORMED AS FOLLOWS: RC_RD, SR_SR-1
788                           *                           THEN RA_RC, RB_RA, RC_RB, RD_RC, SR_SR+1
789                           *
790       0577  30637777237   CAB       RD    ADD         RC    DCSR        ADJUST SR: RC_RD
791       0600  31217757757             RC    ADD         PUSH  CCA   NEXT  TOS_RC, CCA, NEXT
792                           *
793                           *
794                           *         LDXA, ADXA, STAX, ADAX
795                           *     ENTER WITH SR>=1 FOR ALL EXCEPT LDXA
796                           *
797       0601  37762261737   LDXA            JSB   PSHM              SR4     EMPTY A TOS REG IF NEC
798       0602  37217757746         X         ADD         PUSH  CCA   NEXT  TOS_X, CCA, NEXT
799                           *
800       0603  33675757766   ADXA  X     RA    ADDO        RA          NEXT  (S)_(S)+X: CCA,OVFL
801       0604  33557757557   STAX      RA    ADD         X     POPA  NEXT  X_(S), CCA, S_S-1
```

```
      ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:05 PM

  802  0605  33555757566     ADAx X    RA   ADDO      X    POP  NEXT      X_(S)+X; CCA,OVFL; S_S-1
  803                        *
  804                        *
  805                        *          LDXB, ADXB, STBX, ADBX
  806                        *     ENTER WITH SR>=2
  807                        *
  808  0606  37657757746     LDXn X         ADD       RB   CCA  NEXT      (S-1)_X, CCA
  809  0607  32655757766     ADXn X    RB   ADDO      RB        NEXT      (S-1)_(S-1)+X; CCA,OVFL
  810  0610  32557757757     STBy      RB   ADD       X    CCA  NEXT      X_(S-1), CCA
  811  0611  32555757766     ADBx X    RB   ADDO      X         NEXT      X_(S-1)+X; CCA,OVFL
  812                        *
  813                        *
  814                        *          ADD, SUB, NEG, CMP
  815                        *     ENTER WITH SR>=1 FOR NEG, ELSE SR>=2
  816                        *
  817  0612  33655757572     ADD  RB   RA   ADDO      RB   POP  NEXT      (S-1)_(S-1)+(S), S_S-1
  818  0613  33645757572     SUB  RB   RA   SUBO      RB   POP  NEXT      (S-1)_(S-1)-(S), S_S-1
  819  0614  33665757777     NEG       RA   SUBO      RA        NEXT      (S) _ -(S); CCA,OVFL
  820                        *
  821  0615  33767477552     CMP  RB   RA   SUB                 POPA NOFL      CCA ON (S-1)-(S)
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
  822  0616  00773357552          RB   CIR  IOR                 POPA NEXT      REVERSE CCL, CCG IF OVFL
  823  0617  37777757577               ADD                      POP  NEXT        (CIR>0 SO NO CCE IF OVFL)
  824                        *
  825                        *
  826                        *          DADD, DSUB, DNEG, DCMP, ZROB
  827                        *     ENTER WITH SR>=2 FOR DNEG AND ZROB, ELSE SR=4.
  828                        *     FCMP EXITS THROUGH DCM2; ZROB USES A FREE LINE IN DCMP.
  829                        *
  830  0620  33677517231     DADn RC   RA   ADD       RA   DCSR NCRY      (S)_(S-2)+(S)
  831  0621  32654767230          RD   RB   INCO      RB   DCSR UNC       (S-1)_(S-3)+(S-1)+1 IF CRRY
  832  0622  32655777230          RD   RB   ADDO      RB   DCSR           ELSE (S-1)_(S-3)+(S-1)
  833  0623  37766360563                    JMP  DCCA             UNC       (S-2),(S-3) DELETED; SET CC
  834                        *
  835  0624  33667517231     DSUB RC   RA   SUB       RA   DCSR NCRY      (S)_(S-2)-(S)
  836  0625  32645767230          RD   RB   SUBO      RB   DCSR UNC       (S-1)_(S-3)-(S-1) IF CRRY
  837  0626  32645377230          RD   RB   CADO      RB   DCSR           ELSE (S-1)_(S-3)-(S-1)-1
  838  0627  37766360563                    JMP  DCCA             UNC       (S-2),(S-3) DELETED; SET CC
  839                        *
  840  0630  33667517777     DNEG      RA   SUB       RA        NCRY      (S)_-(S)
  841  0631  32645767777               RB   SUBO      RB        UNC       (S-1)_-(S-1) IF CRRY
  842  0632  32645377777               RB   CADO      RB                  ELSE (S-1)_-(S-1)-1
  843  0633  37766360563                    JMP  DCCA             UNC       SET CC
  844                        *
  845  0634  33767477050     DCMP RD   RB   SUB                 CLSR NOFL      CLSR; IF OVFL CCA ON (S-3),
  846  0635  00773357750          RD   CIR  IOR                 CCA  NEXT      (CIR>0 SO NO CCE) DONE
  847  0636  32767417750     DCMB RD   RB   SUB                 CCA  NZRO      CCA ON (S-3)-(S-1), ZERO?
  848  0637  33767517651          RC   RA   SUB                 CCZ  NCRY      YES, CCZ IF (S-2)>=(S)
  849  0640  37657757777     ZROn           ADD            RB        NEXT      RB_0 FOR ZROB
  850  0641  37777757677                    ADD                 CCL  NEXT      ELSE CCL
  851                        *
  852                        *
  853                        *          LCMP, DDEL, DEL, DELB
  854                        *     ENTER WITH SR>=1 FOR DEL, ELSE SR>=2.
  855                        *     STOR MAY EXIT THROUGH DEL; MOVE INSTRS MAY EXIT THROUGH DDEL.
```

```
     ADDRESS   CONTENTS     LABEL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:05 PM

856                                *
857            0642  33767507652   LCMP RB   RA   SUB             CCZ  CRRY      CCZ IF (S-1)>=(S)
858            0643  37777777677                  ADD             CCL           ELSE CCL
859                                *
860            0644  37777777577   DDEL           ADD             POP           S_S-1
861            0645  37777757577   DEL            ADD             POP  NEXT     S_S-1, NEXT
862                                *
863            0646  33657757577   DELA      RA   ADD       RB    POP  NEXT     (S-1)_(S), S_S-1, NEXT
864                                *
865                                *
866                                *         LADD, LSUB, NOT
867                                *    ENTER WTH SR>=1 FOR NOT, ELSE SR>=2
868                                *
869            0647  33657767152   LADD RB   RA   ADD       RB    CTF  UNC      (S-1)_(S-1)+(S), F1=CRRY
870            0650  33647777152   LSUB RB   RA   SUB       RB    CTF           (S-1)_(S-1)-(S), F1=CRRY
871            0651  16777557557              UBUS ADD             POPA NF1      CCA ON (S-1), S_S-1
872            0652  37777757517                   ADD             SCRY NEXT     SCRY IF F1
873            0653  37777757537                   ADD             CCRY NEXT     ELSE CCRY IF NF1
874                                *
875            0654  33667357757   NOT        RA   CAD       RA    CCA  NEXT     (S) = 1'S COMPL (S), CCA
876                                *
877                                *
878                                *         LMPY
879                                *    ENTER WITH SR>=2
880                                *
881            0655  37537777533   LMPY RA         ADD       SP3   CCRY          SP3_(S), CCRY
882            0656  37772607777                   REPN                  20
883            0657  16774333272              RB   UBUS MPAD SR1         INCT CTRM     (S)*(S-1)
884            0660  17657407757                   SBUS ADD       RB    CCA  ZERO     (S-1)_MSW, CCA, ZERO?
885            0661  25677757517                   SP3  ADD       RA    SCRY NEXT     NO: (S)_LSW, SCRY, DONE
886            0662  25677757657                   SP3  ADD       RA    CCZ  NEXT     ELSE (S)_LSW, CCZ, DONE
887                                *
888                                *
889                                *         LDIV
890                                *    ENTER WITH SR>=3, RC,RB=U  RA=V: W=U/V
891                                *
892            0663  32317777577   LDIV       RB   ADD       SP1   POP            SP1_LSU, S_S-1
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
893            0664  33766003130              RA   JMP  TRP4                ZERO     INTEGER DIV BY ZERO
894            0665  30767517632              RB   RD   SUB             CLO  NCRY     OVFL IF MSU>=V
895            0666  32302300674              RB   JSB  LDV2 SP1              UNC      YES: JMP, SP1_MSU
896            0667  32772577437              RB   REPN                  CF2  21       CF2
897            0670  30764332276         UBUS RD   DVSB SL1             INCT CTRM     MSU,LSU/V
898            0671  37677573765         RBUS      ADD  SR1   RA              NF2      (S)_REMAINDER
899            0672  16671700000              UBUS ROM        RA    100000          RESTORE HIGH BIT FROM F2
900            0673  01657757757              SP1  ADD        RB    CCA  NEXT     (S-1)_W, CCA, DONE
901                                *
902                                *    OVFL: MSU_(0,MSU)MOD V, RESULTING IN QUOTIENT MODULO 2**16
903                                *    AND CORRECT REMAINDER UPON COMPLETION OF LDIV.
904                                *    SP1=MSU: NEW MSU RETURNED IN RB, SP1_LSU.
905                                *
906            0674  37772577617   LDV2       REPN                  SOV  21       SOV
907            0675  30764332276         UBUS RD   DVSB SL1             INCT CTRM     0,MSU/V
908            0676  37657573765         RBUS      ADD  SR1   RB              NF2      MSU_REMAINDER
909            0677  16651700000              UBUS ROM        RB    100000          RESTORE HIGH BIT FROM F2
```

```
                ADDRESS   CONTENTS     LABEL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                    FRI, AUG 13, 1976,  2:06 PM

  910           0700  33317707777                RA   ADD       SP1        RSB       SP1_LSU, RETURN
  911                                    *
  912                                    *
  913                                    *          MPYI, MPYM, MPY, MPYL
  914                                    *     ENTER WITH SR>=1 AND PADD = N = CIR(8:15) FOR MPYI
  915                                    *     ENTER WITH OPND=(E) FOR MPYM
  916                                    *     ENTER WITH SR>=2 FOR MPY AND MPYL
  917                                    *
  918           0701  02177777637      MPY,       PADD ADD       BUS  OPND            OPND_N IF MPYI (ELSE
  919           0702  37762221737      MPYM            JSB  PSHM                SR4       OPND=(E)); INSURE 1<=SR<4
  920           0703  26202201732                OPND JSB  PUL1 PUSH      SR2       TOS_OPND, PROCESS LIKE MPY
  921           0704  37777777417      MPY             ADD                 SF2       SF2 IF MPY, MPYI OR MPYM
  922           0705  33537777317      MPYL       RA   ADD       SP3  HBF            F1_SIGN OF (S)
  923                                    *
  924           0706  37772607637                     REPN                CLO  20      CLO
  925           0707  16774333272           RB   UBUS MPAD SR1            INCT CTRM    (S)*(S-1)
  926           0710  17657557777                     SBUS ADD       RB        NF1     RB_MSW
  927           0711  32647777776           UBUS RB   SUB       RB                    IF (S) NEG W_W-2**16(S-1)
  928           0712  17777527777                     SBUS ADD                 POS
  929           0713  33647777772           RB   RA   SUB       RB                    IF (S-1) NEG W_W-2**16(S)
  930           0714  25666170560                SP3  JMP  DTST RA            NF2     RA_LSW, JMP IF MPYL
  931           0715  32777447777                RB   ADD                     NSME    ELSE SOV IF HIGH 17
  932           0716  33763127772           RB   RA   XOR                     POS       BITS ARE NOT THE SAME
  933           0717  37777777617                     ADD                 SOV
  934           0720  33657757557                RA   ADD       RB   POPA NEXT      DELETE MSW, CCA ON LSW
  935                                    *
  936                                    *
  937                                    *          DIV, DIVL
  938                                    *     FOR DIV ENTER WITH SR>=2, RB=U RA=V; FOR DIVL ENTER WITH SR>=3,
  939                                    *     RC,RB=U RA=V, S_S-1; FOR BOTH (S-1)_U/V=W, (S)_REMAINDER.
  940                                    *
  941                                    *     SP0,SP1_ABS(U), DEL (S-1) IF DIVL, SF1 IF SGN U <> SGN V, V_ABS(V)
  942                                    *
  943           0721  32317527777      DIV        RB   ADD       SP1        POS       SP1_ABS(U)
  944           0722  32307777777                RB   SUB       SP1
  945           0723  37326350732                     JMP  DVL2 SP0        UNC       SP0_ZERO=MSU
  946                                    *
  947           0724  32317777777      DIVL       RB   ADD       SP1                  SP1_LSU
  948           0725  33657777577                RA   ADD       RB   POP            SP0_MSU, DELETE (S-1)=LSU,
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
  949           0726  31326140732                     RC   JMP  DVL2 SP0        POS       JMP IF U POS
  950           0727  16327777777                     UBUS CAD       SP0                  ELSE SP0,SP1_ABS(U)
  951           0730  01307517777                     SP1  SUB       SP1        NCRY
  952           0731  37336777775                SP0       INC       SP0
  953                                    *
  954           0732  33767537777      DVL3       RA   SUB                     NEG       V_ABS(V)
  955           0733  16666003130                     UBUS JMP  TRP4 RA            ZERO      JMP IF INTEGER DIV BY ZERO
  956           0734  33763377312                RB   RA   XOR                 HBF       SF1 IF SGN U <> ORG SGN V
  957                                    *
  958                                    *     (S-1)_SP0,SP1/RA=U/V, (S)_REMAINDER, CHECK SIGNS, OVFL
  959                                    *
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (18)  ***  UBUS ON RBUS OR SL1 MISSING FROM DVSB
  960           0735  33764112615           SP0  RA   DVSB SL1            SOV  NCRY    FIRST DIV SUB, SOV
  961           0736  37777757776           UBUS      ADD                      NEXT      DONE IF MSU>=V
```

```
     ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

 962  0737  37772607776          UBUS       REPN                    20        FINISH DIVIDE
 963  0740  33764332276          UBUS RA    DVSB SL1          INCT CTRM
 964  0741  37677553765          RBUS       ADD  SR1  RA         NF1       (S)_REMAINDER
 965  0742  01307777777               SP1   SUB       SP1                   W_ -W IF SGN U <> SGN V
 966  0743  33763127772          RB   RA    XOR                     POS       IF SGN U <> SGN REM
 967  0744  33667777777               RA    SUB       RA                      THEN (S)_ -REM
***  WARNING (12)  ***  ZERO,NZRO,NSME SKIP TESTS MADE ON T-BUS
 968  0745  01777407337               SP1   ADD                FHB  ZERO    CLO IF W=ZERO OR
 969  0746  01763137776          UBUS SP1   XOR                NEG          SGN W=(SGN U XOR SGN V)
 970  0747  37777777637                    ADD                CLO
 971  0750  01657757757               SP1   ADD        RB    CCA  NEXT    (S-1)_W, CCA, DONE
 972                              *
 973                              *
 974                              *         LDI, LDNI
 975                              *    PADD = +-N = CIR(8:15)
 976                              *
 977  0751  37762221737  LDI              JSB  PSHM                SR4       EMPTY A TOS REG IF NEC
 978  0752  02217757757               PADD ADD       PUSH CCA  NEXT       TOS _ +-N, CCA
 979                              *
 980                              *
 981                              *         LDXI, LDXN, ADXI, SBXI
 982                              *    PADD = +-N = CIR(8:15)
 983                              *
 984  0753  02557757777  LDX,        PADD ADD        X         NEXT       X _ +-N
 985  0754  02557757746  ADX, X      PADD ADD        Y    CCA  NEXT       X _ X +-N, CCA
 986                              *
 987                              *
 988                              *         ORI, XORI, ANDI
 989                              *    ENTER WITH SR>=1, PADD = N = CIR(8:15)
 990                              *
 991  0755  33673357744  ORI  PADD RA   IOR        RA   CCA  NEXT       (S)_(S) OR N, CCA
 992  0756  33663357744  XOR, PADD RA   XOR        RA   CCA  NEXT       (S)_(S) XOR N, CCA
 993  0757  33663757744  AND, PADD RA   AND        RA   CCA  NEXT       (S)_(S) AND N, CCA
 994                              *
 995                              *
 996                              *         ADDI, SUBI
 997                              *    ENTER WITH SR>=1, PADD = +-N = CIR(8:15)
 998                              *
 999  0760  33675757764  ADD, PADD RA   ADDO       RA        NEXT       (S) _ (S) +-N; CCA,OVFL
1000                              *
1001                              *
1002                              *         DUP, DDUP
1003                              *    ENTER WITH SR>=1 FOR DUP, SR>=2 FOR DDUP
1004                              *
1005  0761  37762221737  DUP              JSB  PSHM                SR4       EMPTY A TOS REG IF NEC
1006  0762  33217757757               RA   ADD        PUSH CCA  NEXT       TOS_(S), CCA, NEXT
1007                              *
1008  0763  32722221737  DDUP     RB   JSB  PSHM SP2       SR4       EMPTY A TOS REG IF NEC
1009  0764  33177677637               RA   ADD        BUS  OPND SRLS    SP2,OPND_(S-1),(S)
1010  0765  37762261737                    JSB  PSHM                UNC       EMPTY A TOS REG IF NEC
1011  0766  37766360150                    JMP  LDD2                UNC       TOS_SP2,OPND, SET CC
1012                              *
1013                              *
1014                              *         ZERO, DZRO, ZROX
1015                              *
```

	ADDRESS	CONTENTS	LABEL	RBUS	SBUS	FUNC	SHFT	STOR	SPEC	SKIP	COMMENTS
1016	0767	377662/0772	DZR0			JMP	DZR2			SRL3	
1017	0770	37762221737				JSB	PSHM			SR4	MAKE AT LEAST
1018	0771	37762361737				JSB	PSHM			UNC	2 TOS EMPTY
1019	0772	372063600774	DZR2			JMP	ZER2	PUSH		UNC	TOS_0
1020	0773	37762221737	ZER0			JSB	PSHM			SR4	EMPTY A TOS REG IF NEC
1021	0774	37217757777	ZER2			ADD		PUSH		NEXT	TOS_0, NEXT
1022			*								
1023	0775	37557757777	ZRO.			ADD		X		NEXT	X_0, NEXT

```
      ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

1024                                 &              SECTOR 2
1025                                 *
1026                                 *          FP INSTRS: FADD, FSUB, FMPY, FDIV, FNEG,
1027                                 *                     FCMP, FLT , DFLT, FIXR, FIXT
1028                                 *
1029                                 *          SHIFT INSTRS: ASL , LSL , CSL , ASR , LSR , CSR ,
1030                                 *                        DASL, DLSL, DCSL, DASR, DLSR, DCSR,
1031                                 *                        TASL, TASR, TNSL, QASL/QASR
1032                                 *
1033                                 *          STACKOPS: TEST, BTST
1034                                 *
1035                                 *
1036                                 *          FADD, FSUB, FMPY, FDIV
1037                                 *     ENTER WITH SR=4; RD,RC=U  RB,RA=V; (S-3),(S-2)_W = U (OP) V, S_S-2
1038                                 *
1039                                 &1000
1040      1000  32651700000     FSUB       RB   ROM        RB   100000          COMPL SIGN OF V IF FSUB
1041      1001  30321677777     FADD       RD   ROMN       SP0  077777          SP0_ABS(MSU)
1042                                 *
1043                                 *     EXCHANGE U AND V IF NECESSARY SO THAT ABS(U) >= ABS(V)
1044                                 *
1045      1002  32303777625           RBUS RB   AND        SP1  CLO             SP1_ABS(MSV), CLO
1046      1003  16767417155           SP0  UBUS SUB             CTF  NZRO       SF1 IF ABS(MSU)>=ABS(MSV)
1047      1004  33767777151           RC   RA   SUB             CTF               IF EQUAL F1 IF LSU>=LSV
1048      1005  33733157774           SP1  RA   IOR        SP2       NF1         SP2=0 IF ABS(V)=0; JMP,
1049      1006  30526391011                RD   JMP  FAD1 SP3       UNC           SP3_MSU IF ABS(U)>=ABS(V)
1050      1007  31733377255           SP0  RC   IOR        SP2  INCN            SP2=0 IF ABS(U)=0,
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
1051      1010  32537777257                RB   ADD        SP3  INCN              SP3_MSV, EXCHANGE U AND V
1052      1011  35766001171     FAD1       SP2  JMP  UANS            ZERO        W=U IF ABS(V)=0
1053                                 *
1054                                 *     UNPACK U INTO SP2,RD,RC, V INTO SP1,RB,RA, SF1 IF SIGNS DIFFERENT
1055                                 *
1056      1012  34763377310     UNPK RD    RB   XOR                  HBF         SF1 IF SIGNS DIFFERENT
1057      1013  30721677700                RD   ROMN       SP2  077700          SP2_EXPU
1058      1014  32303777765           RBUS RB   AND        SP1                  SP1_EXPV
1059      1015  30601600077                RD   ROMN       RD   000077          DELETE SIGN,EXP FROM MSU
1060      1016  32643777765           RBUS RB   AND        RB                   DELETE SIGN,EXP FROM MSV
1061      1017  30611600100                RD   ROM        RD   000100          SET LEADING ONE IN MSU
1062      1020  32657707765           RBUS RB   ADD        RB        RSB         AND IN MSV, RETURN IF JSB
1063                                 *
1064                                 *     W_U IF EXPU-EXPV>24 ELSE ALIGN V WITH U, ADD INTO RB,SP3,SP1
1065                                 *
1066      1021  35367377774           SP1  SP2  CAD        CTRH                 CTR_EXPV-EXPU-1
1067      1022  16771523100                UBUS ROM                  3100 POS    DIFF>24? (ABS(U)>=ABS(V))
1068      1023  37766391171                     JMP  UANS                 UNC    YES, W=U
1069      1024  25777547317                SP3  ADD                  HBF  F1     F1_SIGN OF U=SIGN OF W
1070      1025  33526461031                RA   JMP  FAD4 SP3       UNC         JMP,SP3_LSV IF U,V SAME SGN
1071      1026  33527517777                RA   SUB        SP3       NCRY        ELSE SP3_ -LSV
1072      1027  32647767777                RB   SUB        RB        UNC           IF CRRY MSV_ -MSV
1073      1030  32647377777                RB   CAD        RB                      ELSE MSV_ -MSV-1
1074      1031  35331637600     FAD4       SP2  ROM        SP0  037600          EXPW_EXPU+254 (FOR PACK)
1075      1032  37317777777                     ADD        SP1                  SP1_GUARD BITS=0 IF EXPU=V
1076      1033  32772337277                RB   REPC             INCT CTRM    EXPONENTS EQUAL? NO; ALIGN
```

```
      ADDRESS    CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

1077          1034  37640733276        UBUS       QASR SR1  RB   INCT CTRM       V WITH J; SR RB,SP3,SP1
1078          1035  25537517231        RC   SP3   ADD        SP3  DCSR NCRY      DCSR, SP3_LSU+LSV
1079          1036  32776777777             RB    INC
1080          1037  16657417230        RD   UBUS  ADD        RB   DCSR NZRO      DCSR (=2), RB_MSU+MSV+CRRY
1081          1040  25666001200             SP3   JMP  FNG2 RA        ZERO      JMP, RA_0 IF W=RB,SP3=0
1082                                *
1083                                *    NORMAL EXIT FOR FADD,FSUB,FMPY,FDIV,FLT,DFLT.
1084                                *    NORMALIZE RB,SP3,SP1 TO MSW(8); SHIFT CNT ADDED TO CTR.
1085                                *
1086          1041  32772067777   NORM       RB    REPC                    BIT8      NORMALIZED TO MSW(8)?
1087          1042  16640002277              UBUS  QASL SL1  RB   INCT BIT8      NO; SL RB,SP3,SP1 UNTIL BT8
1088                                *
1089                                *    ROUND AND PACK NORMALIZED W=RB,SP3 INTO RB,RA; TEST FOR UN/OVFL.
1090                                *    SR>=2, SP0=EXP+256-2, CTR=DELTA EXP FRM NORMALIZATION, F1 IF W NEG.
1091                                *
1092          1043  25536517777             SP3   INC        SP3       NCRY      ROUND LSW
1093          1044  32656777777             RB    INC        RB                  INCR MSW IF CRRY
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
1094          1045  37640553716        UBUS       QASR SR1  RB   CCG  NF1       SR1 W=RB,SP3 AFTER ROUNDING
1095          1046  37777777677                   ADD             CCL            CCG IF NF1 ELSE CCL
1096          1047  15767777775        SP0  CTRH  SUB
1097          1050  16771740200              UBUS  ROM                 140200    EXP=EXP-NORM SHFT CNT-256+2
1098          1051  32657777336        UBUS  RB   ADD        RB   FHB           RB_MSW+EXP, SET SIGN,
1099                                *                                            +1 TO EXP FROM NORM, POSS-
1100                                *                                            IBLY +1 MORE FROM ROUNDING
1101          1052  17675022305        RBUS  SBUS CRS  SL1  RA   HBF  EVEN      RA_CSL(ABS(MSW+EXP)),
1102                                *                                              SF1 IF UNFL
1103          1053  25666301056             SP3   JMP  FOV  RA        UNC       JMP, RA_LSW IF UN/OVFL
1104          1054  25773007473        RA   SP3   IOR             SF1  ZERO      UNFL IF ABS(W)=0
1105          1055  25677757777             SP3   ADD        RA        NEXT      ELSE RA_LSW, DONE
1106          1056  37526153132   FOV             JMP  TRP2 SP3       NF1       SP3_0; OVFL IF NF1 (2C JMP)
1107          1057  25666353131             SP3   JMP  TRP3 RA        UNC       ELSE RA_LSW, UNDERFLOW
1108                                *
1109                                *
1110                                *         FMPY
1111                                *    CHECK FOR ZERO, UNPACK U AND V
1112                                *
1113          1060  31533007630   FMPY RD   RC    IOR        SP3  CLO  ZERO      CLO
1114          1061  33533017772        RB   RA    IOR        SP3       NZRO      IF U OR V=0; SP3,RC_W=0
1115          1062  37626301171                   JMP  UANS RC        UNC         CLSR, PUSH W, SET CCE
1116          1063  31522301012             RC    JSB  UNPK SP3       UNC       SP3_LSU; UNPACK U AND V
1117                                *                                              SF1 (W NEG) IF SIGNS DIFF
1118          1064  35337777774        SP1  SP2   ADD        SP0                 SP0_EXPV+EXPU=EXPW+256
1119                                *
1120                                *    MPY MSU,LSU * MSV,LSV = SUM OF THE FOUR PARTIAL PRODUCTS
1121                                *    SHOWN BELOW.  IN GENERAL SUM OF PREVIOUS PARTIAL PRODUCTS
1122                                *    ADDED (AS INTERMEDIATE PRODUCT) INTO MPAD CALCULATING THE NEXT
1123                                *    PARTIAL PRODUCT.  THIS IS DONE IN SUCH A WAY THAT, AND SINCE
1124                                *    MSU AND MSV ARE 7 BITS, OVERFLOW BEYOND AND BETWEEN THE 8 AND 16
1125                                *    BIT ENTITIES USED TO HOLD THE SUM OF PARTIAL PRODUCTS CANNOT OCCUR,
1126                                *    AND MSW IS <= 6 BITS.
1127                                *
1128                                *                    MS1 MS1 LS1 LS1  LSU*LSV
1129                                *          +    MS2 MS2 LS2           MSV*LSU
1130                                *          +    MS3 MS3 LS3           MSU*LSV
```

```
1131                                 *          + MS4 LS4                          MSU*MSV
1132                                 *          = MSW LSW LSW  G                   (G=GUARD BITS)
1133                                 *
1134          1065  37772607777                 REPN                    20
1135          1066  16774333273          RA   UBUS MPAD SR1          INCT CTRM      P1=LSU*LSV
1136          1067  17737777777               SBUS ADD        SP2                   SP2,SP3 _ P1
1137          1070  32537777777               RB   ADD        SP3                   SP3_MSV (BITS(0:8) ARE 0)
1138          1071  35772707777               SP2  REPN                   10
1139          1072  16774333271          RC   UBUS MPAD SR1          INCT CTRM      P2=MSV*LSU+SR8(MSP1)
1140          1073  17317777777               SBUS ADD        SP1                   SP1,SP3(0:7) _ P2
1141          1074  30537777777               RD   ADD        SP3                   SP3_MSU (BITS(0:8) ARE 0)
1142          1075  25772700777               SP3  REPN LRZ               10        SR8(LSP2)
1143          1076  16774333273          RA   UBUS MPAD SR1          INCT CTRM      P3=MSU*LSV+SR16(LSP2)+MSP2
1144          1077  17737777774          SP1  SBUS ADD        SP2                   SP2_MSP3=SP1+MSP2
1145          1100  25317777777               SP3  ADD        SP1                   SP1(0:7)_LSP3=W GUARD BITS
1146          1101  30537777777               RD   ADD        SP3                   SP3_MSU (BITS(0:8) ARE 0)
1147          1102  35772700777               SP2  REPN LRZ               10        ALIGN LH MSP3 WITH MSU,V

1148          1103  16774333272          RB   UBUS MPAD SR1          INCT CTRM      P4=MSU*MSV+SR8(LH MSP3)
1149          1104  17657777777               SBUS ADD        RB                    RB(10-11:15)_MSP4=MSW
1150          1105  35777774237               SP2  ADD  RRZ        DCSR             SP3_LSW; LH LSW=LSP4
1151          1106  25537777236          UBUS SP3  ADD        SP3  DCSR               =SP3(0:7), RH LSW=RH MSP3
1152          1107  37766361041                    JMP  NORM                UNC       NORMALIZE, RND AND PACK
1153                                 *          SR=2 SP0=EXP+256 RB,SP3,SP1=W/4 CTR=0, <4 OF 8 BITS IN SP1 REQ
1154                                 *
1155                                 *
1156                                 *          FDIV
1157                                 *     CHECK FOR ZERO, UNPACK U AND V
1158                                 *
1159          1110  31533377630    FDIV RD   RC   IOR        SP3  CLO                CLO; IF U=0 SP3,RC=0=W
1160          1111  32773017053          RA   RB   IOR             CLSR NZRO        CLSR; V=0?
1161          1112  37766361166               JMP  FDZR                UNC       YES, FP DIV BY ZERO
1162          1113  25766001171               SP3  JMP  UANS           ZERO      IF U=0 W=0, PUSH W, SET CCE
1163          1114  33537771777               RA   ADD  LLZ  SP3                   LH SP3_LH LSV
1164          1115  37762361012                    JSB  UNPK                UNC       UNPACK U AND V,
1165          1116  33677775217               RA   ADD  RLZ  RA   INSR               SF1 (W NEG) IF SIGNS DIFF
1166          1117  25657776212          RB   SP3  ADD  SWAB RB   INSR               SHIFT V LEFT 8, SR_2
1167          1120  35771677500               SP2  ROM                  077500       SP0_EXPU-EXPV+512-3
1168          1121  01327777776          UBUS SP1  SUB        SP0                     =EXPV(BIASED +256) +256-3
1169          1122  31317775777               RC   ADD  RLZ  SP1
1170          1123  31777771777               RC   ADD  LLZ                         SHIFT U LEFT 8 INTO SP2,SP1
1171                                 *
1172                                 *     U = SP2,SP1 = 0 1XX X X X X   X X X X00 0 0
1173                                 *     V = RB,RA   = 0 1XX X X X X   X X X X00 0 0
1174                                 *
1175                                 *     CALC Q1 = U/MSV CARRIED OUT 11 PLACES,
1176                                 *          R1 = REMAINDER - Q1*LSV; IF R1<0 THEN R1_R1+V, Q1_Q1-1.
1177                                 *
1178                                 *     Q1   = 0 0 01X X X XXX
1179                                 *        OR 0 0 001 X X XXX
1180                                 *     R1                  0X XX XXX X X xXX
1181                                 *     Q1*LSV              00 00 0XX X X xXX    X X X X00 0 0
1182                                 *     V IF ADD BACK       01 XX XXX X X xXX    X X X X00 0 0
1183                                 *
1184          1124  16732656770          RD   UBUS REPN SWAB SP2          13
1185          1125  32764332276          UBUS RB   DVSB SL1.         INCT CTRM      SP1_Q1=U/MSV
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

1186             1126   37617773765           RBUS       ADD  SR1  RD                      RD_R1
1187             1127   33537770777                RA    ADD  LRZ  SP3                     (LH) SP3_LSV
1188             1130   37772707777                      REPN                    10
1189             1131   16774333274           SP1  UBUS MPAD SR1        INCT CTRM        Q1*LSV
1190             1132   17527377770           RD   SBUS CAD       SP3                     SP3_MSR1=R1-MS(Q1*LSV)-1
1191             1133   25627417777                SP3  SUB       RC          NZRO        RC_LSR1 = -LS(Q1*LSV)
1192             1134   25536777777                SP3  INC       SP3                     IF LSR1=0 THEN MSR1_MSR1+1
1193             1135   16777537777                UBUS ADD                     NEG         R1<0?
1194             1136   01606301143                SP1  JMP  FDV2 RD          UNC         NO; JMP, RD_Q1
1195             1137   37607377774           SP1        CAD       RD                      ELSE RD_Q1-1
1196             1140   31637517773           RA   RC    ADD       RC          NCRY        LSR1_LSV+LSR1
1197             1141   25536767772           RB   SP3   INC       SP3         UNC           IF CRRY MSR1_MSV+MSR1+1
1198             1142   25537777772           RB   SP3   ADD       SP3                       ELSE MSR1_MSV+MSR1
1199                                  *
1200                                  *     Q1 = RD      = 0 0 0xx X X XXX
1201                                  *     R1 = SP3,RC =                0X XX X X X X    X X X X00 0 0
1202                                  *     V  = RB,RA  =                01 XX X X X X    X X X X00 0 0
1203                                  *
1204                                  *     CALC Q2 = R1/MSV CARRIED OUT 15 PLACES; MS Q2 BT ALIGNS WITH LS Q1.
1205                                  *     IF MS Q2 (Y) = 0 THEN R2 = REMAINDER - Q2*LSV, LEFT JUSTIFY
1206                                  *       LS 14 BITS OF Q2, IF R2<0 THEN Q2_Q2-1;
1207                                  *     ELSE Q2_1'S, RESULTING IN OVFL INTO Q1 AFTER ROUNDING.
1208                                  *
1209                                  *     Q2 = 0 YXX X X X xxx
1210                                  *     R2 =                0X XX X X X X
1211                                  *     Q2*LSV              00 XX X X X X    X X X X00 0 0
1212                                  *
1213             1143   31317777437   FDV2        RC   ADD       SP1  CF2                 CF2 (SHOULD BE 0),
1214             1144   25772617777                SP3  REPN                    17          SP1_LSR1
1215             1145   32764332276           UBUS RB   DVSB SL1        INCT CTRM        SP1_Q2=R1/MSV
1216             1146   37657773765           RBUS       ADD  SR1  RB                      RB_R2
1217             1147   31531777774                      ROM       SP3  177774            OVFL TO Q1 IF MS Q2=1,
1218             1150   01637522777                SP1  ADD  SL1  RC          POS           SP3_Q2=1'S (RND WILL
1219             1151   37766301163                     JMP  FDV3                UNC           INCR Q2 CARRYING INTO Q1)
1220             1152   33537770777                RA   ADD  LR7  SP3                     (LH) SP3_LSV
1221             1153   37772707777                      REPN                    10
1222             1154   16774333274           SP1  UBUS MPAD SR1        INCT CTRM        Q2*LSV
1223             1155   17647377772           RB   SBUS CAD       RB                      RB_MSR2=R2-MS(Q2*LSV)-1
1224             1156   25767417777                SP3  SUB                      NZRO        LSR2 = -LS(Q2*LSV)
1225             1157   32656777777                RB   INC       RB                      IF LSR2=0 THEN MSR2_MSR2+1
1226             1160   31537772777                RC   ADD  SL1  SP3                     SP3 _ (LEFT JUSTIFIED) Q2
1227             1161   32777527777                RB   ADD                      POS         SP3(14:15)=0,
1228             1162   25531777777                SP3  ROM       SP3  177777              IF R2<0 Q2_Q2-1
1229                                  *
1230                                  *     SHIFT Q1,Q2 RIGHT 3 INTO RB,SP3,SP1 = W*2.
1231                                  *     RD=Q1, SP3=Q2, SP0=EXP+256-3, CTR=0, F1=SIGN, <=14 MS Q2 BITS REQ.
1232                                  *
1233             1163   30772757777   FDV3        RD   REPN                    03
1234             1164   37640733276           UBUS       QASR SR1  RB   INCT CTRM        SR3 Q1,Q2 INTO RB,SP3,SP1
1235             1165   37766301041                      JMP  NORM                UNC         NORMALIZE, RND AND PACK;
1236                                  *                                                       <2 MS SP1 BITS REQ
1237                                  *
1238                                  *     FP DIV BY ZERO; PUSH W=U, SET CC, JMP TO TRP5; OVFL,SR CLR
1239                                  *
1240             1166   30217417757   FDZ0        RD   ADD       PUSH CCA  NZRO        PUSH MSW, SET CCA
```

```
     ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
1241      1167  31777777657                RC   ADD             CCZ          IF MSW=0 SET CCZ ON LSW
1242      1170  30206303127                RD   JMP  TRP5 PUSH       UNC     PUSH LSW, TRP5
1243                                   *
1244                                   *    UANS USED BY FADD,FSUB,FMPY,FDIV AS EXIT IN CERTAIN CASES.
1245                                   *    CLSR, W_0 IF -0, PUSH W, SET CC) W=SP3,RC (NORMALLY 0), OVFL CLR
1246                                   *
1247      1171  25777772057  UANS          SP3  ADD  SL1        CLSR         CLSR
1248      1172  16213007771            RC   UBUS IOR       PUSH      ZERO    ABS(U)=0? YES, MSW_0
1249      1173  25677777777                SP3  ADD       RA                  NO, MSW_MSW
1250      1174  30206210563                RD   JMP  DCCA PUSH       SRN7    LSW_LSW, SET CC (2C JMP)
1251                                   *
1252                                   *
1253                                   *         FNEG
1254                                   *    ENTER WITH SR>=2, RB,RA=V; V _ -V,
1255                                   *    FADD,FSUB,FLT,DFLT,FIXR,FIXT EXIT THROUGH FNG2 IF W=0.
1256                                   *
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
1257      1175  32651700000  FNEG         RB   ROM         RB   100000         TOGGLE SIGN
1258      1176  32773007773            RA   RB   IOR                 ZERO    WAS V ZERO?
1259      1177  00773357752            RB   CIR  IOR                 CCA NEXT NO; SET CCA (CIR SO NO CCE)
1260      1200  37657757737  FNG2            ADD       RB   CCE NEXT       YES; REPLACE MSW=0, SET CCE
1261                                   *
1262                                   *
1263                                   *         FCMP
1264                                   *    ENTER WITH SR=4, RB,RC=U RB,RA=V; COMPARE U WITH V, S_S-4
1265                                   *
1266      1201  32763127050  FCMP RD   RB   XOR                 CLSR POS     CLSR; SAME SIGN?
1267      1202  00773357750            RD   CIR  IOR                 CCA NEXT NO; CCA ON U(CIR SO NO CCE)
1268      1203  30766140636                RD   JMP  DCM2              POS     YES; CMP U WITH V IF POS
1269      1204  37777777257                     ADD                 INCN         ELSE SWAP U AND V
1270      1205  37777777257                     ADD                 INCN         TO CMP V WITH U
1271      1206  37766340636                     JMP  DCM2                UNC
1272                                   *
1273                                   *
1274                                   *         FLT, DFLT
1275                                   *    ENTER WITH SR>=1; (S),TOS_FLOAT((S))
1276                                   *    ENTER WITH SR>=2; (S-1),(S)_FLOAT((S-1),(S))
1277                                   *
1278      1207  37331702400  FLT             ROM       SP0  102400         SP0_SNGLE EXP BASE FOR NORM
1279      1210  37762241737                   JSB  PSHM                SR4     EMPTY A TOS REG IF NEC
1280      1211  37217767777                   ADD       PUSH      UNC     PUSH LSW=0
1281      1212  37331704400  DFLT             ROM       SP0  104400         IF DFLT SP0_DBL EXP BASE
1282                                   *
1283                                   *    NORMALIZE, ROUND AND PACK W = RB,SP3,SP1 _ 0;ABS(RB,RA)
1284                                   *
1285      1213  32537417317                RB   ADD       SP3  HBF  NZRO       SP3_MSW, F1_SIGN
1286      1214  33766001200                RA   JMP  FNG2              ZERO    JMP IF W=0 (SET CCE)
1287      1215  37646151221                     JMP  DFL2 RB              NF1     RB_0; JMP IF W POS
1288      1216  33667517777                RA   SUB       PA                  NCRY    ELSE LSW_ -LSW
1289      1217  25527767777                SP3  SUB       SP3         UNC       IF CRRY MSW_ -MSW
1290      1220  25527377777                SP3  CAD       SP3                   ELSE MSW_ -MSW-1
1291      1221  33306301041  DFL2          RA   JMP  NORM SP1         UNC     SP1_LSW; NORM,RND,PACK
1292                                   *
1293                                   *
```

```
1294                                *          FIXR, FIXT
1295                                *     ENTER WITH SR>=2, RB,RA=V; RB,RA_FIX(V), ROUNDED OR TRUNCATED
1296                                *
1297          1222  37777777417     FIXT            ADD             SF2             SF2 IF FIXT
1298                                *
1299                                *     SAVE SIGN, MASK EXP, DETERMINE WHICH DIRECTION TO ADJUST FRACTION
1300                                *
1301          1223  32775777317     FIXR      RB    ADDO            4BF             F1_SIGN; CLO, CCRY
1302          1224  33537777777           RA    ADD       SP3                       SP3_LSV
1303          1225  32761600077           RB    ROMN            000077            DELETE EXP FROM MSV
1304          1226  16651600100           UBUS  ROM       RB    000100              AND ADD ASSUMED BIT
1305          1227  32761677700           RB    ROMN            077700
1306          1230  16371735100           UBUS  ROM       CTRH  135100            EXP-256-23
1307          1231  16726121247           UBUS  JMP  FIX4 SP2       POS           SHIFT LEFT IF EXP-256>=23
1308                                *
1309                                *     EXP-256<23; ADJUST FRACTION RIGHT; CTR=EXP-256-23
1310                                *
1311          1232  16771523000           UBUS  ROM             3000 POS           IF EXP-256 < -1 THEN
1312          1233  37666301200                 JMP  FNG2 RA        UNC             JMP, (S),(S-1)_0, SET CCE
1313          1234  32312337277           RB    REPC      SP1   INCT CTRM          SR V=RB,SP3,SP1(0:8)
1314          1235  37640733276     UBUS        QASR SR1  RB    INCT CTRM          ABS(EXP-256-22) BITS
1315          1236  01766121242           SP1   JMP  FIX2                POS       NO RND IF MS GUARD BIT=0
1316          1237  37766121242                 JMP  FIX2                F2        NO RND IF FIXT INSTR
1317          1240  25536517777           SP3   INC       SP3               NCRY    ROUND LSW
1318          1241  32656777777           RB    INC       RB                        INCR MSW IF CRRY
1319                                *
1320                                *     COMPLEMENT ANSW IF V NEG, SET/CLR CRRY, SET CC.
1321                                *     RB=MSW, SP3=LSW, F1=SIGN V.
1322                                *
1323          1242  25666130560     FIXP      SP3   JMP  DTST RA        NF1       RA_LSW; JMP IF W POS
1324          1243  25667517777           SP3   SUB       RA        NCRY    RA_ -LSW
1325          1244  32647767777           RB    SUB       RB        UNC       IF CRRY MSW_ -MSW
1326          1245  32647377777           RB    CAD       RB                  ELSE MSW_ -MSW-1
1327          1246  37766360560                 JMP  DTST             UNC
1328                                *
1329                                *     EXP-256>=23; ADJUST FRACTION LEFT; SP2=EXP-256-23.
1330                                *     VARIOUS INSTRS ENTER AT TRIE IF INTEGER OVFL DETECTED.
1331                                *
1332          1247  35367377777     FIX4      SP2   CAD       CTRH              -(EXP-256-23)-1
1333          1250  16771521000           UBUS  ROM             1000 POS          INTEG OVFL IF EXP-256>30
1334          1251  37526003133     TRIE        JMP  TRP1 SP3       ZERO          SP3_0, 2C+ JMP
1335          1252  32312377777           RB    REPC      SP1               SHIFT V=RB,SP3,SP1(0:8)
1336          1253  16640332277           UBUS  QASL SL1  RB    INCT CTRM          LEFT (EXP-256)-22 BITS
1337          1254  37766361242                 JMP  FIX2             UNC       CHECK SIGN
1338                                *
1339                                *
1340                                *          ASR, LSR, CSR, TEST, BTST
1341                                *     ENTER WITH SR>=1; FOR SHIFTS PADD=CIR(10:15)
1342                                *
1343          1255  02777777767     SHFR XC   PADD ADD
1344          1256  16347377777           UBUS  CAD       CTRL              CTR _ -CNT-1
1345          1257  33772337277           RA    REPC            INCT CTRM
1346          1260  16676333277           UBUS  CTSS SR1  RA    INCT CTRM          SHIFT IF CNT NZRO
1347                                *
1348          1261  33777757757     TEST      RA    ADD             CCA   NEXT      CCA, NEXT
```

```
      ADDRESS   CONTENTS      LAB  RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

1349                          *
1350        1262  33777754017 BTST       RA   ADD  RRZ        CCB  NEXT      CCB ON (S)(8:15), NEXT
1351                          *
1352                          *
1353                          *         ASL, LSL, CSL
1354                          *    ENTER WITH SR>=1; PADD=CIR(10:15)
1355                          *
1356        1263  02777777767 SHF1 XC   PADD ADD
1357        1264  16347377777           UBUS CAD        CTRL                    CTR _ -CNT-1
1358        1265  33772337277           RA   REPC             INCT CTRM
1359        1266  16676332277           UBUS CTSS SL1  RA   INCT CTRM      SHIFT IF CNT NZRO
1360        1267  33777757757           RA   ADD              CCA  NEXT      CCA, NEXT
1361                          *
1362                          *
1363                          *         DASL, DLSL, DCSL
1364                          *    ENTER WITH SR>=2, PADD=CIR(10:15)
1365                          *
1366        1270  02777777767 SHD1 XC   PADD ADD
1367        1271  16347377777           UBUS CAD        CTRL                    CTR _ -COUNT-1
1368        1272  33306330563           RA   JMP  DCCA SP1        CTRM      EXIT IF COUNT=ZERO
1369        1273  32772377277           RB   REPC             INCT            TBUS _ MSW
1370        1274  16653732277           UBUS CTSD SL1  RB   INCT CTRM      SHIFT LEFT
1371        1275  01666330563           SP1  JMP  DCCA RA         UNC       SET CC ON RESULT
1372                          *
1373                          *
1374                          *         DASR, DLSR, DCSR
1375                          *    ENTER WITH SR>=2, PADD=CIR(10:15)
1376                          *
1377        1276  02777777767 SHD2 XC   PADD ADD
1378        1277  16347377777           UBUS CAD        CTRL                    CTR _ -COUNT-1
1379        1300  33526330563           RA   JMP  DCCA SP3        CTRM      EXIT IF COUNT=ZERO
1380        1301  32772377277           RB   REPC             INCT            TBUS _ MSW
1381        1302  16653733277           UBUS CTSD SR1  RB   INCT CTRM      SHIFT RIGHT
1382        1303  25666330563           SP3  JMP  DCCA RA         UNC       SET CC ON RESULT
1383                          *
1384                          *
1385                          *         TASL, TASR
1386                          *    ENTER WITH SR>=3, PADD=CIR(10:15)
1387                          *
1388        1304  02777777767 TAS1 XC   PADD ADD
1389        1305  16347377777           UBUS CAD        CTRL                    CTR _ -COUNT-1
1390        1306  33306331313           RA   JMP  TAS2 SP1        CTRM      SP1_W(32:47); EXIT IF CNT=0
1391        1307  32537707777           RB   ADD       SP3        RSB       SP3_W(16:31); RET IF TASR
1392        1310  31772377277           RC   REPC             INCT
1393        1311  16620332277           UBUS QASL SL1  RC   INCT CTRM      SL RC,SP3,SP1
1394        1312  25657777777           SP3  ADD       RB                    RB_W(16:31)
1395                          *
1396        1313  31777407757 TAS2       RC   ADD              CCA  ZERO      CCA ON MSW; ZERO?
1397        1314  01677757777           SP1  ADD       RA         NEXT      NO; RA_W(32:47), DONE
1398        1315  01773377652      RB   SP1  IOR              CCZ            YES; CCZ ON W(16:47),
1399        1316  01677757777           SP1  ADD       RA         NEXT        RA_W(32:47), DONE
1400                          *
1401                          *
1402        1317  37762361304 TASR            JSB  TASL              UNC       SET UP SP3,SP1,CTR FOR SHFT
1403        1320  31772377277           RC   REPC             INCT
```

```
1404          1321  37620733276          UBUS       QASR SR1  RC    INCT CTRM      SR RC,SP3,SP1
1405          1322  25646301313               SP3   JMP  TAS2 RB          UNC       RB_W(16:31), SET CC
1406                                    *
1407                                    *
1408                                    *         TNSL
1409                                    *    ENTER WITH SR>=3, RC,RB,RA=W, PADD=CIR(10:15)
1410                                    *    IF W(6:47)<>0 THEN W(0:5)_0, NORMALIZE LEFT TO BIT6, X_XC+SHIFT CNT
1411                                    *    ELSE X_XC+42 (W UNCHANGED)
1412                                    *
1413          1323  31721601777     TNS1       RC    ROMN       SP2  001777        SP2_W(0:15) WITH W(0:5)=0
1414          1324  32526011330               RB    JMP  TNS2  SP3        NZRO      SP3_W(16:31)=ZERO?
1415          1325  37311600052                     ROM        SP1  000052        YES, SP1_42
1416          1326  35773017773          RA   SP2   IOR                   NZRO      W(6:15,32:47) ALSO ZERO?
1417          1327  01557757727          XC   SP1   ADD        X     CCE  NEXT      YES; X_XC+42, CCE, DONE
1418          1330  33317777777     TNS2       RA    ADD        SP1                 SP1_W(32:47)
1419          1331  35632057777               SP2   REPC       RC          BIT6      RC_W(0:15) WITH W(0:5)=0,
1420          1332  16620052277               UBUS  QASL SL1  RC    INCT BIT6        IF NO BIT6 SL SP2,SP3,SP1
1421          1333  25657777777               SP3   ADD        RB                     INTO RC,RB,SP1 UNTIL BIT6
1422          1334  01677777777               SP1   ADD        RA                  RA_W(32:47)
1423          1335  14557757707          XC   CTRL  ADD        X     CCG  NEXT      X_XC+SHIFT CNT, CCG, DONE
1424                                    *
1425                                    *
1426                                    *         QASL/QASR
1427                                    *    ENTER WITH SR=4, PADD=CIR(10:15)
1428                                    *
1429          1336  02777777766     QAL0 X     PADD ADD
1430          1337  16347377777               UBUS CAD        CTRL                CTR _ -COUNT-1
1431          1340  32306301346               RB   JMP  QLR3 SP1        CTRM      SP1_W(32:47); EXIT IF CNT=0
1432          1341  31526241353               RC   JMP  QLR5 SP3        INDR      SP3_W(16:31); JMP IF QASR
1433          1342  30772377277               RD   REPC                  INCT
1434          1343  16600332273          RA   UBUS  QASL SL1  RD    INCT CTRM      SL RD,SP3,SP1,RA
1435          1344  37677777765          RBUS       ADD        RA                  RA_W(48:63)
1436                                    *
1437          1345  25637777777     QLR3       SP3   ADD        RC                  RC_W(16:31)
1438          1346  30777407757     QLR4       RD    ADD                   CCA  ZERO      CCA ON MSW; ZERO?
1439          1347  01657757777               SP1   ADD        RB          NEXT      NO; RB_W(32:47), DONE
1440          1350  01657777777               SP1   ADD        RB                    YES; RB_W(32:47),
1441          1351  01773377771          RC   SP1   IOR                                 CCZ ON W(16:63),
1442          1352  33773357656          UBUS RA    IOR                   CCZ  NEXT      DONE
1443                                    *
1444          1353  30772377277     QLR5       RD    REPC                  INCT
1445          1354  33600733276               UBUS  RA   QASR SR1  RD    INCT CTRM      SR RD,SP3,SP1,RA
1446          1355  17666301345                     SBUS JMP  QLR2 RA          UNC       RA_W(48:63), SET CC
```

```
                ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

1447                                   &              SECTOR 3
1448                                   *
1449                                   *          FIELD AND BIT INSTRS: EXF , DPF , SCAN,
1450                                   *                                TRBC, TSBC, TCBC, TBC
1451                                   *
1452                                   *          PSHR, SETR, XCHD, ADDS, SUBS
1453                                   *
1454                                   *          XEQ, LLSH/RSW, OPTX
1455                                   *
1456                                   *          I/O INSTRS: SIO, RIO, WIO, TIO, CIO, SIN, CMD, SED,
1457                                   *                      RMSK/RCLK, SMSK/SCLK
1458                                   *
1459                                   *          SUBROUTINES: AS-K, IOPD/A, PULL, PSHM, PSHA, BNDC
1460                                   *
1461                                   *
1462                                   *          EXF, DPF
1463                                   *     ENTER WITH SR>=1 FOR EXF, SR>=2 FOR DPF; PADD=CIR(8:15)
1464                                   *
1465                                   &1400
1466          1400  02377772764      DEXF PADD PADD ADD  SL1  CTRH                 CTR _ J
1467          1401  02531377760           PADD ROMI       SP3  177760             SP3 _ -(16-K)
1468          1402  14537777764      PADD CTRL ADD        SP3                     SP3(12:15) _ J+K MOD 16
1469          1403  25352377777           SP3  REPC       CTRL                    CTR _ -(16-K)
1470          1404  16337733277           UBUS ADD  SR1    SP0  INCT CTRM         SP0_K BITS RIGHT JUSTIFIED
1471          1405  25351377760           SP3  ROMI       CTRL 177760             CTR _ -(16-(J+K MOD 16))
1472          1406  00766001412           CIR  JMP  DPF                 BITS      JMP IF DPF
1473                                   *
1474          1407  33772377777      EXF       RA   REPC                          ROTATE (S) SO K BITS
1475          1410  16735333277           UBUS CRS  SR1    SP2  INCT CTRM           ARE RIGHT JUSTIFIED
1476          1411  35663757755      SP0  SP2  AND         RA   CCA  NEXT         (S)_K BITS FROM (S), CCA
1477                                   *
1478          1412  33663777775      DPF  SP0  RA   AND         RA                  (S)=FIELD TO BE
1479          1413  16772377777           UBUS REPC                                 DEPOSITED IN S-1
1480          1414  16675332277           UBUS CRS  SL1    RA   INCT CTRM         ALIGN (S) WITH (S-1)
1481          1415  25351377760           SP3  ROMI       CTRL 177760             CTR _ -(16-(J+K MOD 16))
1482          1416  37772377775      SP0       REPC
1483          1417  16735332277           UBUS CRS  SL1    SP2  INCT CTRM         ALIGN MASK WITH (S-1)
1484          1420  35762777772      RB   SP2  CAND                              SAVE GOOD DATA
1485          1421  16653357553      RA   UBUS IOR         RB   POPA NEXT         DEPOSIT FIELD, CCA
1486                                   *
1487                                   *
1488                                   *          SCAN
1489                                   *     ENTER WITH SR>=1
1490                                   *
1491          1422  33766001430      SCAN      RA   JMP  SCN2            ZERO      JMP IF (S)=0
1492          1423  33772137777           RA   REPC                      NEG       SHIFT LEFT
1493          1424  16677532277           UBUS ADD  SL1    RA   INCT NEG           LOOKING FOR A ONE
1494          1425  14556647766      X    CTRL INC         X              INDR      X_X+CNT+1; CIR(4)?
1495          1426  14557777777           CTRL ADD         X                        NO, X_CNT

1496          1427  33677757753           RA   RA   ADD         RA   CCA  NEXT      SHIFT ONE MORE, CCA
1497                                   *
1498          1430  37771600020      SCN2           ROM                 000020      (S)=0;
1499          1431  16557757727           XC   UBUS ADD         X    CCE  NEXT        X_XC+16, CCE
1500                                   *
```

```
1501                                    *
1502                                    *         TRBC, TSBC, TCBC, TBC
1503                                    *     ENTER WITH SR>=1, PADD=CIR(10:15)
1504                                    *
1505          1432  37762361440         TRBC                JSB  TBC              UNC       TEST BIT, SET CC
1506          1433  35662757773               RA    SP2     CAND       RA         NEXT      RESET BIT
1507                                    *
1508                                    *
1509          1434  37762361440         TSBC                JSB  TBC              UNC       TEST BIT, SET CC
1510          1435  35673357773               RA    SP2     IOR        RA         NEXT      SET BIT
1511                                    *
1512                                    *
1513          1436  37762361440         TCBC                JSB  TBC              UNC       TEST BIT, SET CC
1514          1437  35663357773               RA    SP2     XOR        RA         NEXT      COMPLEMENT BIT
1515                                    *
1516                                    *
1517          1440  02777777467         TBC   XC    PADD    ADD                SF1
1518          1441  16347374337               UBUS  CAD     RRZ   CTRL FHB               CTR = (-N-1)MOD 64
1519          1442  16732331277               UBUS  REPC    LLZ   SP2  INCT CTRM          SP2=UBUS=100000
1520          1443  16735333277               UBUS  CRS     SR1   SP2  INCT CTRM          SP2 = SHIFT (N)MOD 64
1521          1444  35763707753               RA    SP2     AND                CCA  RSB     SET CC; RETURN IF NOT TBC
1522          1445  37777757777                             ADD                     NEXT      NEXT IF TBC
1523                                    *
1524                                    *
1525                                    *         PSHR
1526                                    *     PUSH DS,DQ,X,STA,DZ,DDL,DB-BANK,DB,S-BANK
1527                                    *     AS SPECIFIED BY CIR(15:8); DB-BANK,DB,S-BANK ARE PRV
1528                                    *     PADD=CIR(8:15)
1529                                    *
1530          1446  21302211744         PSHB        Q     JSB   PSHA SP1             SRN7      SP1=Q; EMPTY TOS REGS
1531          1447  23771600011               SM    ROM                000011          SM+9
1532          1450  16767777142               Z     UBUS  SUB                CTF               F1 IF Z>=SM+9 (Z<64K-9)
1533          1451  23326151752               SM    JMP   BND2 SP0               NF1       STOV(BND2) IF NF1; SP0=SM
1534          1452  02735123777               PADD  CRS   SR1   SP2               POS       PUSH DS IF CIR(15)
1535          1453  22207777775               SP0   DB    SUB        PUSH
1536          1454  02307055317               PADD  CAD   RLZ   SP1  HBF  BIT6          LH SP1=NOT CIR(8:15),F1=NOT
1537                                    *                                                   CIR(8);PUSH DQ IF CIR(14)
1538          1455  22207777774               SP1   DB    SUB        PUSH
1539          1456  35357423777               SP2   ADD   SR1   CTRL               EVEN      CTR=CIR(8:13);
1540                                    *                                                   PUSH X IF CIR(13)
1541          1457  37217777766               X           ADD        PUSH
1542          1460  14357423777               CTRL  ADD   SR1   CTRL               EVEN      PUSH STA (AND EMPTY TOS
1543          1461  24202211744               STA   JSB   PSHA PUSH               SRN7        IF SR>1) IF CIR(12)
1544                                    *     AT THIS POINT SR<=3
1545          1462  14357423777               CTRL  ADD   SR1   CTRL               EVEN      PUSH DZ IF CIR(11)
1546          1463  22207777762               Z     DB    SUB        PUSH
1547          1464  34322211744               DL    JSB   PSHA SP0               SRNZ      SP0=DL; EMPTY TOS REGS
1548          1465  01775132374               SP1   SP1   CRS   SL1         LBF  NEG       SF2 IF NOT CIR(9);
1549                                    *                                                   PUSH DDL IF CIR(10)
1550          1466  22207777775               SP0   DB    SUB        PUSH
1551          1467  02761410300               PADD  ROMN                0300 NZRO          PUSH DB-BNK,DB(NF2) OR
1552                                    *                                                   S-BNK(NF1) IF CIR(8 OR 9)
1553          1470  37777757777         PSHB                ADD                     NEXT      ELSE DONE
1554          1471  37766253117                             JMP   TRP6              NPRV      YES; MODE VIOL IF NPRV
1555          1472  03217577417               RBR   ADD         PUSH DB   NF2               PUSH DB-BANK
```

```
ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

 1473   03677757617              RBR  ADD        RA   S    NEXT      SUBST S-BANK, DONE IF F2
 1474   22206141470              DB   JMP  PSH2 PUSH      F1        PUSH DB; DONE IF F1
 1475   03217757617              RBR  ADD        PUSH S    NEXT      PUSH S-BANK; DONE
                          *
                          *
                          *        SETR
                          *    SET S-BANK,DB,DB-BANK,DL,Z,STA,X,Q,S
                          *    AS SPECIF BY CIR(8:15); S-BNK THROUGH Z AND STA(0,1,3,8:15) ARE PRV
                          *    ENTER WITH SR=4, PADD=CIR(8:15)
                          *
 1476   02311777760    SETB      PADD ROM             SP1  177760      SP1 POS IF ANY CIR(8:11)
 1477   03537667617              RBR  ADD        SP3  S    NPRV      SP3_S-BANK
 1500   01346361507              SP1  JMP  SET1 CTRL      UNC       CTR_CIR(12:15); JMP IF PRV
                          *
 1501   01346123117              SP1  JMP  TRP6 CTRL      POS       ELSE TRP6 IF ANY CIR(8:11)
 1502   33321627400              RA   ROMN       SP0  027400        ISOLATE (S)(2,4:7)
 1503   24301760377              STA  ROMN       SP1  150377        SP1_STA WITH BITS 2,4:7 CLR
 1504   15367057777              CTRH CAD        CTRH      BIT6      SET STA(2,4:7) IF CIR(12)
 1505   01513377575         SP0  SP1  IOR        STA  POP
 1506   37766361526              JMP  SET3                 UNC       CHECK CIR(13:15)
                          *
 1507   02317525777    SET1      PADD ADD  RL7  SP1            POS       IF CIR(8) THEN
 1510   33537777577              RA   ADD       SP3  POP                  SP3_(S) (SET S-BANK)
 1511   01317532577              SP1  ADD  SL1  SP1  POP  NEG       SET DB,DB-BNK IF CIR(9)
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
 1512   33206361515              RA   JMP  SET2 PUSH       UNC       NO, REPLACE TOS
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
 1513   30457777577              RD   ADD       DB   POP             YES; SET DB,DB-BANK
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
 1514   33157777417              RA   ADD       SBR  DB
 1515   01317522777    SET9      SP1  ADD  SL1  SP1            POS       SET DL IF CIR(10)
 1516   22717777573         RA   DB   ADD       DL   POP
 1517   37762201732              JSB  PUL1                 SRZ       FILL AT LEAST ONE TOS REG
 1520   01317522777              SP1  ADD  SL1  SP1            POS       SET Z IF CIR(11)
 1521   22257777573         RA   DB   ADD       Z    POP
 1522   37762201732              JSB  PUL1                 SRZ       FILL AT LEAST ONE TOS REG
 1523   15367057777              CTRH CAD        CTRH      BIT6      SET STA IF CIR(12)
 1524   33517777577              RA   ADD       STA  POP
 1525   37762201732              JSB  PUL1                 SRZ       FILL AT LEAST ONE TOS REG
                          *
 1526   15773063364    SET3 PADD CTRH IOR  SR1       LBF  BIT8      SF2 IF CIR(14); CIR(13)?
 1527   33557777577              RA   ADD       X    POP               YES, SET X
 1530   23726171535              SM   JMP  SET4 SP2       NF2       SP2_SM, JMP IF NO CIR(14)
 1531   33762201732              RA   JSB  PUL1                 SRZ       FILL AT LEAST ONE TOS REG
 1532   22317777776         UBUS DB   ADD       SP1                 SP1_NEW Q
 1533   23722361751              SM   JSB  BNDC SP2            UNC       SP2_SM, CK NEW Q
 1534   01437777577              SP1  ADD       Q    POP               SET Q
 1535   02777437777    SET7      PADD ADD                       ODD       SET S IF CIR(15)
 1536   25157757617    SET5      SP3  ADD       SBR  S    NEXT      ELSE S-BANK_SP3, DONE
 1537   33762201732              RA   JSB  PUL1                 SRZ       FILL AT LEAST ONE TOS REG
 1540   22317777776         UBUS DB   ADD       SP1                 SP1_NEW S
 1541   16722361751              UBUS JSB  BNDC SP2            UNC       SP2_NEW S, CK BOUNDS
 1542   37762211744              JSB  PSHA                 SRNZ      EMPTY TOS REGS
 1543   35466361536              SP2  JMP  SETS SM           UNC       SET S
                          *
```

```
1608                                 *
1609                                 *         XCHD
1610                                 *    ALSO ENTRY FOR DISP, PSDB AND PSEB
1611                                 *    ENTER WITH SR>=2, PADD=CIR(12:15)
1612                                 *
1613          1544  37766263117      XCHD           JMP  TRP6            NPRV     XCHD,DISP,PSDB,PSEB ARE PRV
1614          1545  02726012643           PADD JMP  DISP SP2             NZRO     IF CIR(14,15) DISP, PSDB
1615                                 *                                              OR PSEB INSTRS, SET SP2>0
1616                                 *
1617                                 *    EXCHANGE DB-BANK,DB WITH (S-1),(S)
1618                                 *
1619          1546  33457777777           RA   ADD       DB                       SWITCH RA AND DB
1620          1547  22677777777           DB   ADD       RA
1621          1550  32157777417           RB   ADD       SBR  DB                  SWITCH RB AND DB-BANK
1622          1551  03657757417           RBR  ADD       RB   DB   NEXT
1623                                 *
1624                                 *
1625                                 *         ADDS, SUBS
1626                                 *    ENTER WITH SR>=1, PADD = +-N = CIR(8:15)
1627                                 *
1628          1552  33307367777    SUBS       RA   CAD       SP1        UNC      SUBS ENTRY; SP1 _ -(S)-1
1629          1553  37307377773    ADDS RA         CAD       SP1        UNC      ADDS ENTRY; SP1 _ (S)-1
1630          1554  02302011744           PADD JSB  PSHA SP1             NZRO     SP1-N, EMPTY REGS IF N<>0;
1631          1555  01302211744           SP1  JSB  PSHA SP1             SRNZ     ELSE RESTORE SP1,
1632                                 *                                              AND EMPTY TOS REGS
1633          1556  23316777774           SP1  SM   INC       SP1                 SP1 _ S + SP1 + 1
1634          1557  37762361751                     JSR  BNDC                UNC      BNDC ON SP1, DB<>ZERO
1635          1560  37467357774           SP1       CAD       SM         NEXT     S _ SP1 - 1, NEXT
1636                                 *
1637                                 *
1638                                 *         XEQ
1639                                 *    PADD = K = CIR(12:15)
1640                                 *
1641          1561  02307117761    XEQ  SR   PADD CAD       SP1        NCRY     SR>K, IN TOS REGS?
1642          1562  02317767777           PADD ADD       SP1        UNC      YES; SP1_TOS PTR
1643          1563  23136767714           SP1  SM   INC       BSP0 RNS   UNC      NO; READ FROM MEM
1644          1564  37177767223           MREG       ADD       BUS  NIR   UNC      YES; LD NIR FROM TOS REGS
1645          1565  22766777775           SP0  DB   BNDT                          NO; CHECK BOUNDS IF NPRV
1646          1566  20411777777           P    ROM       P    177777             P_P-1 SO INSTR AFTER
1647          1567  37777757777                     ADD                 NEXT       XEQ FOLLOWS XEQED INSTR
1648                                 *
1649                                 *
1650                                 *         LLSH/RSW
1651                                 *    LINKED LIST SEARCH UNTIL CNT=0, TEST<=TARGET OR TARGET=-1;
1652                                 *      X=CNT, RB,RA=ABS PTR TO LIST, RC=TEST WORD, RD=TARGET OFFSET;
1653                                 *      NEXT BNK,ADDR IN LAST LINK AND TEST WORD IN "NEXT" LINK
1654                                 *      MUST BE IN LEGAL MEMORY.
1655                                 *    READ SWITCH REG; TOS_SWCH
1656                                 *    ENTER WITH SR=4, PADD=CIR(8:15)
1657                                 *
1658          1570  02766021611    LLSW       PADD JMP  RSW              EVEN     JMP IF RSW INSTR
1659          1571  33326263117           RA   JMP  TRP6 SP0             NPRV     SP0_RA; LLSH IS PRV
1660          1572  37556777766           X         INC       X                   INC X FOR LOOP
1661          1573  32157767017           RB   ADD       SBR  ABS   UNC      SET LINK BANK
1662          1574  26157777017    LLS1       OPND ADD       SBR  ABS           SET NEXT LINK BANK
```

```
                ADDRESS   CONTENTS      LABEL  RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS
1663             1575     30177777173          RA   RD   ADD       BUS  ROA           READ TARGET WORD
1664             1576     37547017766          X         CAD       X              NZRO  DECR COUNT; REPLACE LAST
1665             1577     37677757675          SP0       ADD       RA   CCL  NEXT     ADDR, CCL, DONE IF ZERO
1666             1600     33176777177               RA   INC       BUS  ROA           READ NEXT LINK ADDR
1667             1601     26727377151          RC   OPND CAD       SP2  CTF           SF1 IF TEST>TARGET
1668             1602     03657777017               RBR  ADD       RB   ABS           STORE LINK BANK
1669             1603     37766323000                    JMP  IR0                TEST  JMP IF INT PENDING
1670             1604     33137557177               RA   ADD       BSP0 ROA  NF1     READ NEXT LINK BANK
1671             1605     26666361574               OPND JMP  LLS1 RA         UNC   JMP, RA_NEXT ADDR IF F1
1672             1606     35767407771          RC   SP2  SUB                  ZERO  TARGET ALL 1'S?
1673             1607     37777757737                    ADD                 CCE  NEXT  NO, CCE
1674             1610     37777757717                    ADD                 CCG  NEXT  YES, CCG
1675                      *
1676             1611     37762361737          RSW            JSR  PSHM            UNC   EMPTY ONE TOS REG
1677             1612     07217757757               SWCH ADD       PUSH CCA  NEXT  TOS_SWCH, CCA
1678                      *
1679                      *
1680                      *         OPTX
1681                      *    OPTIONAL INSTRS 020400/777; ENTER WITH SR=4, PADD=CIR(8:15)
1682                      *
1683             1613     04766027777  OPTx      CPX1 JMP  UNIM                EVEN  UNIM (TRP7) IF NOT PRESENT
1684             1614     02771403400               PADD ROM                3400 BITB  IF CIR(8:15)<%200
1685             1615     16577773765          RBUS UBUS ADD  SR1  RAR               RAR_3400+CIR(8:15)/2
1686             1616     02571212360               PADD ROMI       RAR  012360      ELSE RAR_12360+CIR(12:15)
1687                      *
1688                      *
1689                      *         SIO, RIO, WIO, TIO, CIO, SIN, CMD
1690                      *    PADD = K = CIR(12:15); ENTER WITH SR>=1 FOR SIO,WIO,CIO,CMD
1691                      *
1692             1617     37762361716  SIO            JSB  AS_K                UNC   LOAD DEV# INTO SP1
1693             1620     16531302400               UBUS ROMI       SP3  102400      FORM TIO CMD (RANK1 RSB)
1694             1621     37762361724                    JSB  IOPA                UNC   SEND TO DEVICE
1695             1622     25766121651               SP3  JMP  DVNR                POS   JMP IF SIO NOT READY
1696             1623     01177772154          SP1  SP1  ADD  SL1  BUS  WRA           (DEV# # 4)_(S)
1697             1624     33177777437               RA   ADD       BUS  DATA          = I/O PROG ADDR
1698             1625     01531301000               SP1  ROMI       SP3  101000      FORM SIO CMD
1699             1626     37762361724                    JSB  IOPA                UNC   SEND TO DEVICE
1700             1627     37777757557                    ADD                 POPA NEXT  S_S-1, SET CCE, DONE
1701                      *    SIO IS MODIFIED BY A PATCH (AFTER SEC 0-7 TO PROPERLY
1702                      *    COMPUTE PARITY), DRT PTR IS READ TO MAKE SURE WRITE HAS STARTED
1703                      *    BEFORE I/O STARTS.  PATCH SHOULD BE MOVED TO END OF SEC 3 IF
1704                      *    SEC 3 IS REPLACED, OR PUT IN LINE IF LUT IS ALSO REPLACED.
1705                      *
1706                      *
1707             1630     37762361716  RIO            JSB  AS_K                UNC   LOAD DEV# INTO SP1
1708             1631     37762221737                    JSB  PSHM                SR4   EMPTY A TOS REG IF NEC
1709             1632     01531302400               SP1  ROMI       SP3  102400      FORM TIO CMD
1710             1633     37762361724                    JSB  IOPA                UNC   SEND TO DEVICE
1711             1634     25777532777               SP3  ADD  SL1                NEG   DEVICE READY?
1712             1635     37766361651                    JMP  DVNR                UNC   JMP IF NOT
1713             1636     01531303400               SP1  ROMI       SP3  103400      FORM RIO CMD
1714             1637     37762361724                    JSB  IOPA                UNC   SEND TO DEVICE
1715             1640     25217757737               SP3  ADD       PUSH CCE  NEXT  TOS_DATA, SET CCE, DONE
1716                      *
1717                      *
```

```
                ADDRESS   CONTENTS        LABEL  RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

1718             1641   33722301716      WIO          RA   JSB  AS-K SP2        UNC      SP2_(S), LOAD DEV# INTO SP1
1719             1642   16531302400                   UBUS ROMI       SP3  102400      FORM TIO CMD (RANK1 RSB)
1720             1643   37762301724                        JSB  IOPA                UNC      SEND TO DEVICE
1721             1644   25777532777                   SP3  ADD  SL1                 NEG      DEVICE READY?
1722             1645   37766301651                        JMP  DVNR                UNC      JMP IF NOT
1723             1646   01531301400                   SP1  ROMI       SP3  101400      SP3_WIO CMD, (S) IN SP2
1724             1647   37762301723                        JSB  IOPD                UNC      SEND BOTH TO DEVICE
1725             1650   37777757557                        ADD                 POPA NEXT     S_S-1, SET CCE, DONE
1726                             *
1727             1651   37762221737      DVNR              JSB  PSHM                SR4      EMPTY A TOS REG IF NEC
1728             1652   25217757717                   SP3  ADD       PUSH CCG  NEXT     TOS_DEV STA, SET CCG, DONE
1729                             *
1730                             *
1731             1653   37762301716      TIO               JSB  AS-K                UNC      LOAD DEV# INTO SP1
1732             1654   37762221737                        JSB  PSHM                SR4      EMPTY A TOS REG IF NEC
1733             1655   01531302400                   SP1  ROMI       SP3  102400      FORM TIO CMD
1734             1656   37762301724                        JSB  IOPA                UNC      SEND TO DEVICE
1735             1657   25217757737                   SP3  ADD       PUSH CCE  NEXT     TOS_DEV STA, SET CCE, DONE
1736                             *
1737                             *
1738             1660   33722301716      CIO          RA   JSB  AS-K SP2        UNC      SP2_(S), LOAD DEV# INTO SP1
1739             1661   16531300400                   UBUS ROMI       SP3  100400      FORM CIO CMD (RANK1 RSB)
1740             1662   37762301723                        JSB  IOPD                UNC      SEND CMD AND (S) TO DEVICE
1741             1663   37777757557                        ADD                 POPA NEXT     S_S-1, SET CCE, DONE
1742                             *
1743                             *
1744             1664   37762301716      SIN               JSB  AS-K                UNC      LOAD DEV# INTO SP1
1745             1665   16531300000                   UBUS ROMI       SP3  100000      FORM INT CMD (RANK1 RSB)
1746             1666   37762301724                        JSB  IOPA                UNC      SEND TO DEVICE
1747             1667   37777757737                        ADD                 CCE  NEXT     SET CCE, DONE
1748                             *
1749                             *
1750             1670   37762301716      CMD               JSB  AS-K                UNC      LOAD MCU MOD#/CMD
1751             1671   16177777037                   UBUS ADD       BUS  CRL               SEND TO MCU CONTROL REG
1752             1672   33177777057                   RA   ADD       BUS  CMD               SEND CMD AND (S)
1753             1673   37777757577                        ADD                 POP  NEXT     S_S-1, DONE
1754                             *
1755                             *
1756                             *         SED
1757                             *    PADD = CIR(12:15)
1758                             *
1759             1674   37766203117      SED               JMP  TRP6                NPRV     SED IS PRV
1760             1675   24501737777                   STA  ROMN       STA  137777          DISABLE EXT INTERRUPTS
1761             1676   02772742764                PADD PADD REPN SL1                 04       UBUS(1)_CIR(15) IN 6C SO
1762             1677   16315333277                   UBUS CRS  SR1  SP1  INCT CTRM          ANY PENDING INT OCCURS
1763                             *                                                            IMMEDIATELY FOLLOWING SED
1764                             *                                                            IF DISABLING INTERRUPTS
1765             1700   24513357774                SP1  STA  IOR       STA       NEXT     ENABLE EXT INTS IF CIR(15)
1766                             *
1767                             *
1768                             *         RMSK/RCLK
1769                             *    PADD = -CIR(12:15)
1770                             *
1771             1701   37762221737      RMSK              JSB  PSHM                SR4      EMPTY A TOS REG IF NEC
1772             1702   02776532764                PADD PADD INC  SL1                 NEG      PADD=0 IF RMSK, ELSE <= -1
```

```
      ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

1773    1703  051767?4176           UBUS MOD  INC  RRZ  BUS  ROA  UNC       READ (7 OR 11) IF RMSK
1774    1704  132177?7777                PCLK ADD            PUSH  NEXT      TOS_PCLK IF RCLK
1775    1705  262177?7777                OPND ADD            PUSH  NEXT      ELSE TOS_(7 OR 11)
1776                        *
1777                        *
1778                        *          SMSK/SCLK
1779                        *     ENTER WITH SR>=1, PADD = -CIR(12:15)
1780                        *
1781    1706  337262?3117   SMSK       RA   JMP  TRP6 SP2        NPRV      SMSK AND SCLK ARE PRV
1782    1707  375317?3000              ROM             SP3  103000        SP3 _ SET MASK CMD
1783    1710  023365?2764         PADD PADD INC  SL1  SP0        POS       IF SCLK THEN
1784    1711  330177?7577              RA   ADD        PCLK POP  NEXT        PCLK_(S), S_S-1, DONE
1785    1712  377623?1723                   JSB  IOPD             UNC       ELSE SEND CMD AND MASK
1786    1713  051767?4155         SP0  MOD  INC  RR7  BUS  WRA                                  TO IOP
1787    1714  331777?7457              RA   ADD        BUS  DPOP             (7 OR 11)_(S), S_S-1
1788    1715  377777?7737                   ADD                  CCE  NEXT    SET CCE, DONE
1789                        *
1790                        *
1791                        *     AS-K TRAPS IF NPRV AND RETURNS WITH RANK1 RSB WITH SP1_(S-K)(8:15).
1792                        *     PADD = K = CIR(12:15); CHECK IS MADE FOR E IN TOS REGS.
1793                        *
1794    1716  023062?3117   AS-K       PADD JMP  TRP6 SP1        NPRV      INSTRS USING AS-K ARE PRV
1795    1717  027671?7761         SR   PADD CAD                  CRRY      SR>K?
1796    1720  231767?7776         UBUS SM   INC       BUS  ROS  UNC       NO, READ (S-K) FROM MEM
1797    1721  373177?4763         MREG      ADD  RRZ  SP1        RSB       YES, (S-K) IN TOS REGS
1798    1722  263177?4777              OPND ADD  RRZ  SP1        RSB
1799                        *
1800                        *     IOPD AND IOPA HANDLE CPU-IOP COMMUNICATIONS.
1801                        *     SP2=DATA (FOR IOPD), SP3=CMD/ADDR; IOD RETURNED IN SP3.
1802                        *     IF I/O TIMEOUT: CCL,NEXT IF CTR=0 ELSE SYSH (IXIT,LOAD,DUMP).
1803                        *
1804    1723  350577?7777   IOPD       SP2  ADD        TOD                 TRANSFER DATA
1805    1724  250377?7777   IOPA       SP3  ADD        IOA                 TRANSFER CMD
1806    1725  125377?7777              IOD  ADD        SP3                 SP3_IOD (UNLD IMM AFTR IOA)
1807    1726  047774?3777              CPX1 ADD  SR1              ODD       I/O TIMEOUT?
1808    1727  377777?7777                   ADD                  RSB       NO; RETURN
1809    1730  147660?2762              CTRL JMP  SYSH             NZRO      YES; SYSH IF CTR NZRO
1810    1731  377777?7677                   ADD             CCL  NEXT      ELSE SET CCL, NEXT
1811                        *
1812                        *     PUL1 PULLS ONE WORD INTO A TOS REG, CHECKING FOR
1813                        *       STACK UNDERFLOW (INIT SM<=DB) BEFORE QUEING THE WORD UP.
1814                        *     RANK1 RSB WITH QUP_PULLED WORD=OPND.
1815                        *
1816    1732  231777?7777   PUL1       SM   ADD        BUS  ROS            READ (SM)
1817    1733  227671?7776         UBUS DB   CAD                  CRRY      SM>DB?
1818    1734  237662?3120              SM   JMP  STUN             NPRV      NO, UNDERFLOW IF NPRV
1819    1735  374673?7776         UBUS      CAD        SM                  DECREMENT SM
1820    1736  262777?7217              OPND ADD        QUP  INSR RSB       QUE MEM UP
1821                        *
1822                        *     PSHM PUSHES ONE TOS REG INTO MEM,
1823                        *     CHECKING INCREMENTED SM FOR STACK OVERFLOW.
1824                        *
1825    1737  231767?7757   PSHM       SM   INC        BUS  WRS            PUSH TOS REG INTO MEM
1826    1740  101777?7437              QDWN ADD        BUS  DATA
1827    1741  237671?7762         Z    SM   CAD                  NCRY      Z>=SM+1
```

```
      ADDRESS   CONTENTS      LAB  RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

1828    1742  23476707237            SM   INC        SM   DCSR RSB       YES: INC SM, DCSR, RET
1829    1743  37306302527                 JMP  EX11 SP1        UNC       NO: SP1_0, STOV(EX11), ALL
1830                              *                                          TOS REGS WILL BE SAVED
1831                              *
1832                              *    PSHA PUSHES ALL TOS REGS INTO MEM,
1833                              *    WITHOUT CHECKING FOR OVERFLOW; ENTER WITH SR>=1
1834                              *
1835    1744  23176777757       PSHA SM   INC        BUS  WRS
1836    1745  10177777437            QDWN ADD        BUS  DATA           PUSH REG
1837    1746  23476657237            SM   INC        SM   DCSR SRL>      INC SM AND DCSR
1838    1747  37766361744                 JMP  PSHA           UNC       PUSH NEXT WORD IF SR WAS >1
1839    1750  37777707777                 ADD                 RSB       ELSE RETURN
1840                              *
1841                              *    STACK BOUNDS TEST ROUTINE
1842                              *    OVFL CHECKED BEFORE UNFL IN CASE BOTH OVFL AND UNFL
1843                              *      OVFL IF OUT OF BOUNDS AND WITHIN 32K OF Z (WRAPPING)
1844                              *    VARIOUS IN-LINE OVFL TESTS ENTER AT BND2 IF OVFL DETECTED
1845                              *
1846    1751  01767527762       BNDC Z    SP1  SUB                 POS       SP1>Z OR WRAP AROUND?
1847    1752  37306302527       BND2       JMP  EX11 SP1        UNC       YES: SP1_0, STOV(EX11)
1848    1753  22767527774            SP1  DB   SUB                 POS       SP1<DB OR WRAP AROUND?
1849    1754  37766203120                 JMP  STUN           NPRV      YES: UNDERFLOW IF NPRV
1850    1755  37777707777                 ADD                 RSB       ELSE RETURN
1851                              *
1852                              *
1853                              *    3L  1764-5,1776 USED BY SIO PATCH, INCL SECOND PARITY SEC 2/3
```

```
1854                                &              SECTOR 4
1855                                *
1856                                *         MOVE INSTRS: MOVE, MVBW, MVB , CMPB, SCU , SCW ,
1857                                *                      MVBL, MVLB, MFDS, MTDS, MDS , MABS
1858                                *
1859                                *         SUBROUTINES: DBWC, DBBC, MVWS, DSEG, D03S/D05S
1860                                *
1861                                *
1862                                *         MOVE
1863                                *     MOVE WORDS DB OR PB REL SOURCE TO DB REL TARGET.
1864                                *     RA=SIGNED COUNT, RB=SOURCE PTR, RC=TARGET PTR.
1865                                *     ENTER WITH SR>=3
1866                                *
1867                                &2000
1868   2000  22317777477  MVW0      DB    ADD         SP1  SF1              SF1 IF DB, SOURCE BASE=DB
1869   2001  33766002202  MVWP      RA    JMP  D031                ZERO     EXIT IF CNT=ZERO
1870   2002  37762221737            JSR   PSHM                     SR4      EXACTLY 3 TOS REGS FILLED
1871                                *
1872   2003  37536777777  MVW1            INC         SP3                   SP3=1 IF CNT POS, ELSE -1;
1873   2004  33777527777            RA    ADD                      POS        OPND_CNT INC/DECR
1874   2005  25527777777            SP3   SUB         SP3                     BY ONE TOWARD ZERO;
1875   2006  16167707633      RA    UBUS  SUB         BUS  OPND RSB          RETURN IF NOT MOVE.
1876                                *
1877   2007  23326142016            SM    JMP  MVW2   SP0            F1     JMP IF DB REL SOURCE
1878   2010  36607777760      PL    PB    SUB         RD                    CHECK PB REL SOURCE BNDS
1879   2011  32764777776      UBUS  RB    UBNT                                ASSUME PL-PB POS
1880   2012  26777777772      RB    OPND  ADD                                 ERR IF PL-PB<REL BEG ADDR
1881   2013  16764777770      RD    UBUS  UBNT                                ERR IF PL-PB<REL END ADDR
1882   2014  03357777217            RBR   ADD         CTRL PB               CTR_PB=BANK
1883   2015  36306362020            PB    JMP  MVW3   SP1            UNC    SOURCE BASE=PB
1884   2016  32602262313  MVW3      RB    JSB  DBWC   RD             NPRV   TEST DB REL SOURCE IF NPRV
1885   2017  03357777417            RBR   ADD         CTRL DB               CTR_DB=BANK.        (2C JMPS)
1886   2020  31602262313  MVW4      RC    JSB  DBWC   RD             NPRV   TEST DB REL TARGET IF NPRV
1887   2021  14157777017  MVW4      CTRL  ADD         SBR  ABS              ABS=BANK_SOURCE BASE BANK
1888   2022  22322362353            DB    JSB  MVWS   SP0            UNC    MOVE WORDS, TARGET BASE=DB
1889   2023  37766323000  MVW5            JMP  IRQ                    TEST   JMP IF INTERRUPT PENDING
1890   2024  37766362202            JMP   D031                        UNC    DONE; DEL FROM STACK
1891                                *
1892                                *
1893                                *         MVBW
1894                                *     MOVE BYTES FROM DB REL SOURCE TO DB REL TARGET WHILE ALPHA
1895                                *     OR WHILE NUMERIC, WITH UPSHIFT OF LOWER CASE ALPHA IF SPECIFIED.
1896                                *     LAST TARGET MUST BE IN LEGAL MEMORY.
1897                                *     RA=SOURCE BYTE PTR, RB=TARGET BYTE PTR.
1898                                *     ENTER WITH SR>=2, PADD=CIR(8:15)
1899                                *
1900   2025  34722221737  MVB4      RB    JSB  PSHM   SP2            SR4    SP2_TARGET BYTE PTR
1901   2026  37536677417            INC         SP3  SF2  SRL3             SP3_1 (DELTA FOR MV LOOP)
1902   2027  37762361737            JSB   PSHM                     UNC    EXACTLY 2 TOS REGS FILLED
1903   2030  23302362321            SM    JSB  DBBC   SP1            UNC    TEST TARGET BYTE PTR
1904   2031  14157777017            CTRL  ADD         SBR  ABS              ABS=BANK_CTR=DB=BANK
1905   2032  33737777777            RA    ADD         SP2                   TEST SOURCE BYTE PTR,
1906   2033  30622362321            RD    JSB  DBBC   RC             UNC      RC_TARGET WORD ADDR
1907   2034  30767507151      RC    RD    SUB                  CTF  CRRY   TARGET ADDR>=SOURCE ADDR
```

```
      ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

1908     2035  30307702774          SP1  RD   SUB  SL1  SP1        UNC       NO, SP1_(SM-SOURCE ADDR)*2
1909     2036  31307772774          SP1  RC   SUB  SL1  SP1                  YES, SP1_(SM-TARG ADDR)*2
1910     2037  02777772764          PADD PADD ADD  SL1
1911     2040  16377542776          UBUS UBUS ADD  SL1  CTRH       F1        CTR_CIR(8:13)
1912     2041  25723767773          RA   SP3  AND       SP2        UNC       SP2=1 IF RH OF
1913     2042  25723777772          RB   SP3  AND       SP2                    MAX(SOURCE,TARGET ADDRS)
1914     2043  01771600002               SP1  ROM                 000002      COUNT=COUNT+2
1915     2044  35207777776          UBUS SP2  SUB       PUSH                 RA_CNT (DECR BY 1 IF RH)
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
1916     2045  30326302066               RD   JMP  MB10 SP0        UNC       SP0_SOURCE ADDR; GO MOVE
1917                           *
1918                           *
1919                           *         MVB/CMPB
1920                           *     MOVE BYTES DB OR PB REL SOURCE TO DB REL TARGET;
1921                           *       LAST+1 SOURCE MUST BE IN LEGAL MEMORY IF CNT NZRO.
1922                           *     COMPARE BYTES DB OR PB REL SOURCE WITH DB REL TARGET;
1923                           *       LAST+1 TARGET MUST BE IN LEGAL MEMORY IF CNT NZRO.
1924                           *     RA=SIGNED COUNT, RB=SOURCE BYTE PTR, RC=TARGET BYTE PTR.
1925                           *     ENTER WITH SR>=3
1926                           *
1927     2046  32737777477  MVBn     RB   ADD       SP2  SF1                 SF1 IF DB SOURCE, SP2_PTR
1928     2047  37322221737  MVBa          JSB  PSHM SP0        SR4          EXACTLY 3 TOS REGS FILLED
1929     2050  33606002065               RA   JMP  MV95 RD       ZERO      EXIT IF CNT=ZERO; (0,0) OR
1930                           *                                              (DB,0) READ, CCE IF CMPB
1931     2051  23302302003               SM   JSB  MVW1 SP1        UNC       GET DELTA, ADJUSTED CNT
1932     2052  37762142321               JSB  DB&C                 F1        IF DB REL TEST STARTING
1933     2053  30326142063               RD   JMP  MV13 SP0        F1          AND ENDING SOURCE ADDRS,
1934                           *                                              SP0_SOURCE STARTING ADDR
1935     2054  03357777217               RBR  ADD       CTRL PB             CTR_PB=BANK IF PB REL
1936     2055  36607777760          PL   PB   SUB       RD                  ASSUME PL=PB POS, <16K
1937     2056  26777773772          RB   OPND ADD  SR1                      PB REL ENDING WORD ADDR
1938     2057  16764777770          RD   UBUS UBNT                         CK PL-PB>=REL END ADDR
1939     2060  32337773777               RB   ADD  SR1  SP0                PB REL STARTING WORD ADDR
1940     2061  16764777770          RD   UBUS UBNT                         CK PL-PB>=REL START ADDR
1941     2062  36337777775          SP0  PB   ADD       SP0                 SP0_PB SOURCE STARTING ADDR
1942     2063  14157777017  MVBa         CTRL ADD       SBR  ABS            ABS=BANK_SOURCE BANK
1943     2064  31722212321               RC   JSB  DB&C SP2        SRNZ      TEST TARGET BOUNDS (2C JMP)
1944     2065  00766002132  MVBc         CIR  JMP  CMP0            BITA      CMPB INSTR IF CIR(8)=1
1945                           *
1946                           *     ENTRY POINT FOR MB20 MOVE BYTES LOOP
1947                           *
1948     2066  31775373777  MB1          RC   CRS  SR1
1949     2067  25763377316          UBUS SP3  XOR            HHF           SF1 IF 1B IN FIRST TARGET
1950                           *
1951                           *     MB20 MOVES BYTES FOR MVB AND MVBW INSTRS, MVB IF VF2,
1952                           *     TESTING FOR INTERRUPTION AND COMPLETION.
1953                           *     RA=COUNT, RB=SOURCE PTR, RC=TARGET PTR,
1954                           *     SP0=SOURCE ADDR, RD=TARGET ADDR, SP3=DELTA.
1955                           *
1956     2070  37177567175  MB2n SP0          ADD       BUS  ROA  F2        READ SOURCE WORD
1957     2071  33766002202               RA   JMP  D031            ZERO      DONE MVB; DEL FROM STACK
1958     2072  32735103777               RB   CRS  SR1  SP2        F2        SP2 NEG IF RH OF SOURCE
1959     2073  37766323000                    JMP  IRQ                 TEST      JMP IF MVB AND INT PENDING
1960     2074  25657427772          RB   SP3  ADD       RB          EVEN      UPDATE SOURCE PTR
1961     2075  26317700777               OPND ADD  LRZ  SP1          UNC       SOURCE WAS LH
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

1962             2076   26317774777                OPND ADD  RRZ  SP1                    SOURCE WAS RH
1963             2077   30177567577                RD   ADD       BUS  ROD  F2         READ TARGET WORD
1964             2100   35766362111                SP2  JMP  MB22                UNC    UBUS_SP2; JMP IF MVB INSTR
1965             2101   15761600100                CTRH ROMN                 000100      ISOLATE UPSHIFT BIT
1966             2102   16763773774          SP1   UBUS AND  SR1                        UPSHIFT IF LOWER ALPHA
1967             2103   16302777014          SP1   UBUS CAND      SP1  CCB              AND UPSHIFT BIT ON
1968             2104   15761600600                CTRH ROMN                 000600      ISOLATE CCF
1969             2105   27763417776          UBUS  CC   AND                      NZRO   CCF=CCB OF SOURCE?
1970             2106   37766362124                     JMP  MB24                UNC    NO, DONE MVBW
1971             2107   33766002127                RA   JMP  MB26                ZERO   ERROR IF COUNT=ZERO
1972             2110   35766322124   MB21          SP2  JMP  MB24                TEST   JMP IF MVBW AND INT PENDING
1973             2111   25763127776   MB22  UBUS    SP3  XOR                      POS    UPDATE SOURCE ADDR IF
1974             2112   25337777775          SP0   SP3  ADD       SP0                    LAST BYTE OF SOURCE WORD
1975             2113   25637427771          RC    SP3  ADD       RC         EVEN        UPDATE TARGET PTR
1976             2114   01317765777          SP1   ADD  RLZ  SP1             UNC        SHIFT SOURCE IF TARG WAS LH
1977             2115   26737761777          OPND  ADD  LLZ  SP2             UNC        TARGET WAS RH
1978             2116   26737774777          OPND  ADD  RRZ  SP2                        TARGET WAS LH
1979             2117   30177777557          RD    ADD       BUS  WRD
1980             2120   35173377434          SP1   SP2  IOR       BUS  DATA           WRITE UPDATED TARGET WORD
1981             2121   25667557473          RA    SP3  SUB       RA   SF1  NF1       UPDATE COUNT
1982             2122   25617777450          RD    SP3  ADD       RD   CF1           UPDATE TARGET ADDR IF BYTE
1983             2123   37766362070                     JMP  MB20                UNC     WAS LAST BYTE OF TARG WORD
1984                                    *
1985                                    *    MVBW INT AND NORM COMPLETION (MVB TERMINATION TESTS IMBEDDED ABOVE)
1986             2124   25647727572   MB24  RB    SP3  SUB       RB   POP  TEST      ADJUST SOURCE PTR AND STK
1987             2125   37766362201                     JMP  D035                UNC    DONE MVBW; DEL FROM STK
1988             2126   37766363000                     JMP  IRQ                 UNC    JMP IF INT PENDING
1989                                    *
1990                                    *    MVBW BOUNDS VIOLATION
1991             2127   24766132110   MB2A          STA  JMP  MB21                NEG    NO ERROR IF PRV
1992             2130   25647777572          RB    SP3  SUB       RB   POP            ADJUST SOURCE PTR AND STK
1993             2131   37346213013                     JMP  BNDV CTRL            SRN7   CTR_0; BNDV INT (2C JMP)
1994                                    *
1995                                    *    ENTRY POINT FOR CMPB COMPARE BYTES LOOP
1996                                    *
1997             2132   30177777577   CMPB          RD   ADD       BUS  ROD            READ FIRST TARGET WORD
1998             2133   31775773737                 RC   CRS  SR1       CCE            SET CCE IN CASE CNT=0
1999             2134   25763777316          UBUS  SP3  XOR                HBF        SF1 IF 1B IN FIRST TARGET
2000                                    *
2001                                    *    CMPB COMPARES BYTES UNTIL NOT EQUAL OR CNT=0 FOR CMPB INSTR.
2002                                    *    IT ENDS WITH CCA SET ON LAST (TARGET MINUS SOURCE) SCANNED IF NOT
2003                                    *    EQUAL, OR WITH CCE SET IF ALL TARGET AND SOURCE BYTES ARE EQUAL.
2004                                    *    RA=SIGNED COUNT, RB=SOURCE BYTE PTR, RC=TARGET BYTE PTR,
2005                                    *    SP0=SOURCE ADDR, RD=TARGET ADDR, SP3=DELTA.
2006                                    *
2007             2135   33766002202   CMPB          RA   JMP  D031                ZERO   IF CNT=0 DONE, DEL FR STK
2008             2136   37766323000                     JMP  IRQ                 TEST   JMP IF INT PENDING
2009             2137   25737427771          RC    SP3  ADD       SP2        EVEN        SP2_UPDATED TARGET PTR
2010             2140   26317760777                OPND ADD  LRZ  SP1             UNC    TARGET WAS LH
2011             2141   26317774777                OPND ADD  RRZ  SP1                    TARGET WAS RH
2012             2142   37177777175          SP0         ADD       BUS  ROA            READ SOURCE WORD
2013             2143   25355152477                SP3  CRS  SL1  CTRL SF1  NF1       CTR(5)_SP3(0)
2014             2144   25617777450          RD    SP3  ADD       RD   CF1           UPDATE TARG ADDR IF F1
2015             2145   14763027772          RB    CTRL XOR                 EVEN        UPDATE SOURCE ADDR IF
2016             2146   25337777775          SP0   SP3  ADD       SP0                    LAST BYTE OF SOURCE WORD
```

```
2017          2147  25657427772          RB    SP3  ADD        RB         EVEN      UPDATE SOURCE PTR
2018          2150  26177750637               OPND ADD  LRZ   BUS  OPND UNC       SOURCE WAS LH
2019          2151  26177774637               OPND ADD  RRZ   BUS  OPND           SOURCE WAS RH
2020          2152  25667777773          RA    SP3  SUB        RA                  UPDATE COUNT
2021          2153  30177777577               RD   ADD        BUS  ROD            READ NEXT TARGET WORD
2022          2154  26767417754          SP1  OPND SUB                CCA  NZRO    CCA ON TARGET-SOURCE BYTES
2023          2155  35626302135               SP2  JMP  CMPR RC       UNC       JMP, RC_TARG PTR, IF EQUAL
2024          2156  25677777773          RA    SP3  ADD        RA                  MOVE COUNT, SOURCE PTR
2025          2157  25647777772          RB    SP3  SUB        RB                    BACK TO UNEQUAL BYTES
2026          2160  37766302202               JMP  D031               UNC       DONE; DEL FROM STACK
2027                                *
2028                                *
2029                                *         SCU, SCW
2030                                *    ALSO ENTRY FOR MFTD AND MTDS
2031                                *    SCAN UNTIL TERMINAL OR TEST BYTE FOUND.
2032                                *    SCAN WHILE STRING=TEST BYTE.
2033                                *    LAST+1 SOURCE MUST BE IN LEGAL MEMORY.
2034                                *    RA=TERMINAL BYTE,TEST BYTE; RB=SOURCE BYTE PTR.
2035                                *    ENTER WITH SR>=2, PADD=CIR(12:15)
2036                                *
2037          2161  37777777477   SCU         ADD                SF1               SF1 IF SCJ OR MFDS
2038          2162  02761400010   SCW    PADD ROMN               0010 ZERO
2039          2163  37766302240               JMP  MFTD               UNC       JMP IF MFDS OR MTDS
2040          2164  32722221737          RB   JSB  PSHM SP2        SR4           SP2_SOURCE BYTE PTR
2041          2165  33537674417          RA   ADD  RRZ  SP3  SF2  SRL3          RH SP3_TEST BYTE, SF2
2042          2166  37762301737               JSB  PSHM               UNC       EXACTLY 2 TOS REGS FILLED
2043          2167  23023022321          SM   JSB  DBBC SP1          UNC       CHECK BOUNDS ON START ADDR
2044          2170  30137777577          RD   ADD       RSP0 ROD               READ FIRST WORD
2045          2171  33617750777          RA   ADD  LRZ  RD           NF1       RH RD_TERMINAL BYTE
2046          2172  37766302212               JMP  SCU1               UNC       JMP IF SCJ
2047                                *
2048                                *    SCAN WHILE LOOP
2049                                *    RB=BYTE PTR, SP0=WORD ADDR, RD=TERMINAL BYTE, SP3=TEST BYTE
2050                                *
2051          2173  37762302217   SCW1        JSB  GSCB               UNC       GET SOURCE BYTE
2052          2174  25763017774          SP1  SP3  XOR                NZRO      SOURCE=TEST BYTE?
2053          2175  35646302173               SP2  JMP  SCW1 RB       UNC       YES; UPDATE BYTE PTR, JMP
2054          2176  30763007514          SP1  RD   XOR          SCRY ZERO      SCRY IF LAST SOURCE =
2055          2177  37777777537               ADD               CCRY             TERMINAL BYTE, ELSE CCPY
2056          2200  01777777017               SP1  ADD          CCB             SET CCB ON LAST BYTE
2057                                *
2058                                *    DELETE FROM STACK; EXIT ROUTINE FOR ALL MOVE INSTRS
2059                                *    D05S: S_S-CIR(13:15), ENTER WITH SR=4 AND PADD=SDEC+8; PRV ONLY
2060                                *    D03S: S_S-CIR(14:15), ENTER WITH SR=2
2061                                *    D031:    "   "   "    ENTER WITH SR>=3
2062                                *
2063          2201  37762301732   D03S        JSB  PUL1               UNC       ENTER WITH SR=2
2064          2202  00775123377   D031   CIR  CRS  SR1        LBF  POS         ENTER WITH SR>=3
2065          2203  37777777577               ADD               POP             S_S-1 IF CIR(15)

2066          2204  37766100644               JMP  DDEL               F2        S_S-2 IF CIR(14)
2067          2205  37777757777               ADD                     NEXT      DONE
2068                                *
2069          2206  02311527764   D05S   PADD ROM        SP1  7764 POS          ENTER WITH SR=4,PADD=SDEC+8
2070          2207  37766302202               JMP  D031               UNC       JMP IF SDEC<4
2071          2210  23777777057          SM   ADD               CLSR            ELSE CLSR, (SP1=SDEC-4)
```

```
      ADDRESS    CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

2072        2211  01467757776        UBUS SP1  SUB       SM        NEXT      S_S-SP1-4, DONE
2073                               *
2074                               *    SCAN UNTIL LOOP
2075                               *    RB=BYTE PTR, SP0=WORD ADDR, RD=TERMINAL BYTE, SP3=TEST BYTE, RC=STA
2076                               *
2077        2212  37762362217    SCU1            JSB  GSCB           UNC       GET SOURCE BYTE
2078        2213  30763007514          SP1  RD    XOR            SCRY ZERO      SOURCE=TERMINAL BYTE?
2079        2214  25763017534          SP1  SP3   XOR            CCRY NZRO      SOURCE=TEST BYTE?
2080        2215  37766362201                JMP  D03S           UNC       YES; DONE, DEL FROM STACK
2081        2216  35646362212           SP2  JMP  SCU1 RB        UNC       NO; RB_UPDATED PTR
2082                               *
2083                               *    GSCB SUPPLIES SOURCE BYTES IN RH SP1 FOR SCU AND SCW INSTRS,
2084                               *    SETS CCB, TESTS FOR INTERRUPTS PENDING AND IF NPRV SOURCE ADDR>SM.
2085                               *    SP0=WORD ADDR, RB=BYTE PTR; UPDATED PTR RETURNED IN SP2.
2086                               *    IF INTERRUPT RC LOADED INTO STA BEFORE TRANSFERING TO IRD.
2087                               *
2088        2217  23767117775    GSCB SP0  SM    CAD                NCRV      ADDR>=SM+1
2089        2220  37346263013          JMP  BNDV CTRL       NPRV      YES; CTR_0, BNDV IF NPRV
2090        2221  32736427777          RB   INC       SP2   EVEN      UPDATE BYTE PTR
2091        2222  26317700777          OPND ADD  LRZ  SP1   RSB       SOURCE WAS LH, RETURN
2092        2223  26317724777          OPND ADD  RRZ  SP1   TEST      SOURCE WAS RH
2093        2224  37136707575          SP0       INC       PSP0 ROD  RSB       READ NEXT WORD
2094        2225  37766363000                JMP  IRD          UNC       JMP IF INTERRUPT PENDING
2095                               *
2096                               *
2097                               *         MVBL, MVLB
2098                               *    MOVE WORDS DB REL SOURCE TO DL REL TARGET.
2099                               *    MOVE WORDS DL REL SOURCE TO DB REL TARGET.
2100                               *    RA=+-CNT, RB=SOURCE PTR, RC=TARGET PTR.
2101                               *    INITIAL ENTRY AT MABS OR MDS;
2102                               *    F2 IF MVBL, SR=4, PADD=CIR(12:15).
2103                               *
2104        2226  33766002202    MVBL      RA   JMP  D031           ZERO      EXIT IF COUNT=ZERO
2105        2227  22302361737               DB   JSB  PSHM SP1       UNC       EXACTLY 3 TOS REGS FILLED
2106        2230  03357567617               RBR  ADD       CTRL S    F2        CTR_S-BANK
2107        2231  34306362021               DL   JMP  MVW4 SP1       UNC       MOVE WORDS IF MVLB
2108        2232  14157777417               CTRL ADD       SBR  DB              DB-BNK_CTR=S-BNK
2109        2233  03357777417               RBR  ADD       CTRL DB              CTR_DB-BANK
2110        2234  16157777017               UBUS ADD       SBR  ABS             ABS (SOURCE) BANK_DB-BANK
2111        2235  34322362353               DL   JSB  MVWS SP0       UNC       MOVE WORDS (DB TO DL)
2112        2236  14157777417               CTRL ADD       SBR  DB              RESTORE DB-BANK
2113        2237  37766362023                    JMP  MVW5           UNC       CHECK INT, DEL FROM STACK
2114                               *
2115                               *
2116                               *         MFDS, MTDS
2117                               *    MOVE WORDS FROM DATA SEG TO STACK
2118                               *    RA=CNT, RB=DSEG (SOURCE) PTR, (S-2)=DSEG#, (S-3)=STACK (TARGET) PTR
2119                               *    MOVE WORDS FROM STACK TO DATA SEG
2120                               *    RA=CNT, RB=STACK (SOURCE) PTR, (S-2)=DSEG (TARGET) PTR, (S-3)=DSEG#
2121                               *    INITIAL ENTRY AT SCU OR SCW, F1 IF MFDS; SR>=2, PADD=CIR(12:15)
2122                               *
2123        2240  31762271732    MFTD      RC   JSB  PUL1           SRL3      FILL 4 TOS REGS
2124        2241  16302231732               UBUS JSB  PUL1 SP1       SRN4      SP1_DSEG# IF MFDS
2125        2242  37176772177                    INC  SL1  BUS  ROA           READ DST PTR
2126        2243  37766263117                    JMP  TRP6           NPRV      MFDS AND MTDS ARE PRV
```

```
ADDRESS CONTENTS    LAB   RBUS SBUS FUNC SHFT STOR SPEC SKIP   COMMENTS

2244  33766002206         RA   JMP  DOSS           ZERO   EXIT IF COUNT=ZERO
2245  37536547777              INC       SP3       F1     SP3_1 (DELTA FOR MV LOOP)
2246  30317777417         RD   ADD       SP1  SF2         IF MTDS SP1_DSEG#, SF2
2247  37762362355              JSB  DSEG           UNC    SET UP DSEG
2250  03357577417         RBR  ADD       CTRL DB   NF2    CTR_DB-BANK
2251  01326362260         SP1  JMP  MTDS SP0       UNC    JMP IF MTDS INSTR
                    *
2252  22337777777   MFDS  DB   ADD       SP0              SP0 (TARGET BASE) _ DB
2253  26157777017   MFD2  OPND ADD       SBR  ABS         ABS (SOURCE) BANK_DSEG BNK
2254  31617777777   MFD3  RC   ADD       RD               SWITCH RC AND RD
2255  30622362353         RD   JSB  MVWS RC        UNC    MOVE WORDS
2256  31617777777         RC   ADD       RD               RESTORE RC AND RD
2257  30626362263         RD   JMP  MTD2 RC        UNC    CHECK INT, DEL FROM STACK
                    *
2260  14157777017   MTDS  CTRL ADD       SBR  ABS         ABS(SOURCE)BNK_CTR=DB-BNK
2261  26157777417         OPND ADD       SBR  DB          DB (TARGET) BANK_DSEG BANK
2262  22302212353         DB   JSB  MVWS SP1       SRNZ   SP1 (SRC BASE)_DB; 2C JMP
2263  14157727417   MTD2  CTRL ADD       SBR  DB   TEST   RESTORE DB-BANK
2264  37766362206              JMP  DOSS           UNC    DONE; DEL FROM STACK
2265  37766363000              JMP  IRQ            UNC    JMP IF INTERRUPT PENDING
                    *
                    *
                    *          MABS, MDS
                    *     ALSO ENTRY FOR MVBL AND MVLB
                    *     MOVE WORDS FROM DATA SEG TO DATA SEG
                    *     RA=+-CNT; RB,RC=SOURCE PTR AND DSEG#; RD,(SM)=TARGET PTR AND DSEG#
                    *     MOVE WORDS FROM ABS ADDR TO ABS ADDR
                    *     RA=+-CNT; RC,RB=SOURCE BANK,ADDR; (SM),RD=TARGET BANK,ADDR
                    *     ENTER WITH SR=4, PADD=CIR(12:15)
                    *
2266  37317777417   MABS       ADD       SP1  SF2         SF2, SP1_0 IF MABS OR MVBL
2267  37766263117   MDS        JMP  TRP6           NPRV   MABS,MVBL,MDS,MVLB ARE PRV
2270  23177777777         SM   ADD       BUS  ROS         READ (S-4)
2271  33777537777         RA   ADD                 NEG    SP3=DELTA FOR MV LOOP
2272  37536767777              INC       SP3       UNC      SP3_1 IF CNT POS
2273  37527377777              CAD       SP3                ELSE SP3_-1
2274  02761410010         PADD ROMN           0010 NZRO
2275  37766362226              JMP  MVBL           UNC    JMP IF MVBL OR MVLB
2276  33766002206         RA   JMP  DOSS           ZERO   EXIT IF COUNT=ZERO
2277  03357567417         RBR  ADD       CTRL DB   F2     CTR_DB-BANK
2300  37766362304              JMP  MDS1           UNC    JMP IF MDS
                    *
2301  31157777017   MAB1  RC   ADD       SBR  ABS         ABS-BANK_SOURCE BANK
2302  26157777417         OPND ADD       SBR  DB          DB-BANK_TARGET BANK
2303  37326362254              JMP  MFD3 SP0       UNC    SP0 (TARGET BASE)_0
                    *
2304  37176772177   MDS1       INC  SL1  BUS  ROA         READ DST PTR
2305  26302362355         OPND JSB  DSEG SP1       UNC    SP1_TARGET DSEG#
2306  16737777777         UBUS ADD       SP2              SP2_TARGET DSEG ADDR
2307  37176772177              INC  SL1  BUS  ROA         READ DST PTR
2310  26157777417         OPND ADD       SBR  DB          DB-BANK_TARGET BANK
2311  31302362355         RC   JSB  DSEG SP1       UNC    SP1_SOURCE DSEG#
2312  35326362253         SP2  JMP  MFD2 SP0       UNC    SP0_TARGET DSEG ADDR
                    *
                    *
```

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

2182                            #     DBWC CHECKS DB REL STARTING AND ENDING WORD ADDRS
2183                            #     AGAINST SM AND DL FOR MOVE INSTR.
2184                            #     SP0=SM, RD=REL STARTING ADDR, OPND=CNT INC/DECR TOWARD ZERO.
2185                            #     DBWC RETURNS WITH A RANK1 RSB.
2186                            #
2187      2313  22617777770     DBWC  RD    DB    ADD        RD                    FORM STARTING WORD ADDR
2188      2314  16766777775           SP0   UBUS  BNDT                           BNDV IF SM NOT>=START ADDR
2189      2315  34766777770           RD    DL    BNDT                           BNDV IF START ADDR NOT>=DL
2190      2316  26617777770           RD    OPND  ADD        RD                    FORM LAST WORD ADDR
2191      2317  16766777775           SP0   UBUS  BNDT                           BNDV IF SM NOT>=LAST ADDR
2192      2320  34766707770           RD    DL    PNDT               RSB         BNDV IF LAST ADDR NOT>=DL
2193                            #
2194                            #     DBBC CHECKS DB REL STARTING, AND ENDING IF NF2, BYTE ADDRS.
2195                            #     ABS STARTING WORD ADDR RETURNED IN RD, DB-BANK RETURNED IN CTR.
2196                            #     SP2=REL BYTE ADDR, SP1=SM, OPND=CNT INC/DECR TOWARD ZERO IF NF2.
2197                            #     IF SPLIT STACK (S-BANK<>DB-BANK, DB>Z, DB<DL)
2198                            #       THEN BYTE ADDR CONVERTED DIRECTLY TO WORD ADDR AND RET;
2199                            #     ELSE IF NOT(DL<=WORD ADDR<=SM) THEN ADD 32K TO WORD ADDR;
2200                            #     IF NOT(DL<=WORD ADDR<=SM) AND NPRV THEN BNDV.
2201                            #     IF NF2 THEN END ADDR=WORD ADDR+SIGN((ADJ CNT+BYTE PTR(15))/2),
2202                            #       SF1 AND IF NOT(DL<=END ADDR<=SM) AND NPRV THEN BNDV.
2203                            #     DBBC MAY RETURN WITH A RANK1 RSB.
2204                            #
2205      2321  35777773777   DBBC        SP2   ADD  SR1
2206      2322  22617777776           UBUS  DB    ADD        RD                    RD_ABS WORD ADDR
2207      2323  03357777417           RBR   ADD        CTRL  DB                    SPLIT STACK IF S-BANK IS
2208      2324  03767417616           UBUS  RBR  SUB              S     NZRO        NOT THE SAME AS DB-BANK
2209      2325  22767507762           Z     DB   SUB                    CRRY        SPLIT STACK IF Z<DB
2210      2326  22777707777                 DB   ADD                    RSB         RETURN IF SPLIT STACK
2211      2327  34767507776           UBUS  DL   SUB                    CRRY        SPLIT STACK IF DB<DL
2212      2330  37777707777                 ADD                         RSB         RETURN IF SPLIT STACK
2213      2331  30767517774           SP1   RD   SUB                    NCRY        SM>=WORD ADDR?
2214      2332  34767507770           RD    DL   SUB                    CRRY        WORD ADDR>=DL?
2215      2333  30611700000           RD    ROM        RD    100000               ADD 32K IF ADDR>SM OR <DL
2216      2334  34766567776           UBUS  DL   BNDT                   F2          BNDV IF NPRV AND
2217      2335  30766667774           SP1   RD   BNDT                   NPRV          ADDR<DL OR ADDR>SM,
2218      2336  30766707774           SP1   RD   BNDT                   RSB           RETURN IF PRV OR F2
2219                            #
2220      2337  35761600001               SP2   ROMN             000001
2221      2340  26777777316           UBUS  OPND ADD              HBF               BYTE PTR(15)+ADJ CNT
2222      2341  16777773337                 UBUS  ADD  SR1        FHB               REL ENDING WORD ADDR
2223      2342  16737777470           RD    UBUS ADD        SP2   SF1               ABS ENDING WORD ADDR, SF1
2224      2343  34766777776           UBUS  DL   BNDT                             BNDV IF NPRV AND
2225      2344  35766707774           SP1   SP2  BNDT                   RSB         ADDR<DL OR ADDR>SM
2226                            #
2227                            #     MVWS MOVES WORDS AND UPDATES PTRS.
2228                            #     SP1=SOURCE BASE (ABS-BANK), SP0=TARGET BASE (DB-BANK),
2229                            #     RA=NZRO CNT, RB=SOURCE PTR, RC=TARGET PTR, SP3=DELTA.
2230                            #     RETURN IF INTERRUPT (RANK1 RSB) OR COUNT=ZERO.
2231                            #
2232      2345  25657777772   MW11  RB    SP3  ADD        RB                    UPDATE SOURCE PTR
2233      2346  25637777771         RC    SP3  ADD        RC                    UPDATE TARGET PTR
2234      2347  31177777555         SP0   RC   ADD        BUS   WRD             STORE AT TARGET ADDR
2235      2350  26177727437               OPND ADD        BUS   DATA TEST
2236      2351  25667417773         RA    SP3  SUB        RA          NZRO      UPDATE CNT; ELSE UPDATE CNT
```

```
      ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

2237    2352  25667707773          RA   SP3  SUB       RA        RSB        AND RETURN IF ZERO OR INT
2238    2353  32177777174    MVW%  SP1  RB   ADD       BUS  ROA            READ WORD FROM SOURCE
2239    2354  37766362345               JMP  MW11           UNC
2240                         *
2241                         *    DSEG SETS UP DATA SEGMENTS FOR MFDS, MTDS
2242                         *    AND MDS INSTRS; CHECKING FOR UNKN OR ABS DSEG.
2243                         *    SP1=DSEG#, OPND=(2)=DST PTR;
2244                         *    SP0,SP1 USED, F1 CLEARED; SP2,SP3 AND CTR UNAFFECTED.
2245                         *    DSEG BANK RETURNED IN OPND, RANK1 RSB WITH SP1_DSEG ADDR.
2246                         *
2247    2355  26137777177    DSE%       OPND ADD       BSP0 ROA            READ DST LENGTH
2248    2356  01777726774          SP1  SP1  ADD  SL1                      DSEG# * 4
2249    2357  16137777175          SP0  UBUS ADD       BSP0 ROA            READ AUR-LENGTH/4
2250    2360  26767117774          SP1  OPND CAD                 NCRY      DSEG# >= DSTL+1
2251    2361  37306363121               JMP  DSTV SP1          UNC       YES; SP1_0, DSTV (2C JMP)
2252    2362  01766003121          SP1  JMP  DSTV              ZERO      ALSO DSTV IF DSEG# = 0
2253    2363  37177777155          SP0       ADD       BUS  WRA            RETURN AUR-LENGTH/4
2254    2364  26171240000               OPND ROMI      BUS  020000           TO MEM WITH R BIT SET
2255    2365  37776726457               INC  SL1       CF1               CF1 IN CASE ABS TRAP
2256    2366  16136777175          SP0  UBUS INC       BSP0 ROA            READ ADDR
2257    2367  26766162372               OPND JMP  DSG2              NEG       JMP IF DSEG ABS
2258    2370  37167377175          SP0       CAD       BUS  ROA            READ BANK
2259    2371  26317707777               OPND ADD       SP1       RSB       SP1_DSEG ADDR; RETURN
2260                         *
2261                         *    F2=0 IF MDS, CTR=0 IF MFTD SO NO INT CONDITION IS CLEARED
2262    2372  37731721001    DSG%       ROM       SP2  121001             ABS DSEG LABEL=34,1
2263    2373  01526363140          SP1  JMP  INT7 SP3       UNC          SP3_DSEG#, TRAP
2264                         *    DSG2 IS MODIFIED BY A PATCH (AFTER SEC 0-7 TO PROPERLY
2265                         *    COMPUTE PARITY).  IN MDS DB-BNK_TARGET BNK BEFORE SECOND
2266                         *    TRIP TO DSEG.  THE PATCH RESTORES DB-BNK FROM CTR IF MDS AND
2267                         *    EITHER DSEG IS ABS, BEFORE EXITING TO INT7.  MDS CAN STILL EXIT
2268                         *    TO DSTV, THROUGH DSEG, WITHOUT RESTORING DB-BNK.  THIS PATCH
2269                         *    SHOULD BE REMOVED IF IT BECOMES NECESSARY TO REPLACE SECTOR 4/5;
2270                         *    AND; CTR (INSTEAD OF DB-BNK)_TARG BNK @2310 AND INSERT 2L AFTER
2271                         *    2311 TO EXCHANGE CTR AND DB-BNK TO COMPLETELY FIX THE PROBLEM.
2272                         *
2273                         *
2274                         *    4L  2374-7 USED BY DSG2 PATCH, INCL SECOND PARITY SEC 4/5
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

2275                                    &              SECTOR 5
2276                                    *
2277                                    *          LLBL, SCAL, PCAL, SXIT, EXIT
2278                                    *
2279                                    *          IXIT/PON/LOCK/UNLK
2280                                    *
2281                                    *          DISP/PSDB/PSEB
2282                                    *
2283                                    *          SUBROUTINES: STMK, CLAB, SSEG, PWR
2284                                    *
2285                                    *          HALT, PAUS INSTRS: STOP, WAIT, SYSH ENTRIES
2286                                    *
2287                                    *
2288                                    *          LLBL
2289                                    *     LOAD LABEL: PADD = N = CIR(8:15)
2290                                    *
2291                                    &2400
2292          2400  37177777360       LLBL  PL        ADD       BUS  ROP             READ STTL AT PL
2293          2401  02537775777             PADD  ADD  RLZ  SP3                     SP3(0:7)=N
2294          2402  24337774777             STA   ADD  RRZ  SP0                     ISOLATE SEG# FROM STA
2295          2403  02722221737             PADD  JSB  PSHM SP2       SR4           SP2=N: EMPTY ONE TOS REG
2296          2404  37302362705                   JSB  CLAB SP1       UNC           SP1=0: CHECK STTL
2297          2405  26777527477             OPND  ADD                 SF1  POS      SF1: EXTERNAL LABEL?
2298          2406  16217757777             UBUS  ADD       PUSH      NEXT          YES: PUSH LABEL, DONE
2299          2407  25766132711             SP3   JMP  CLA2                NEG      STTV(CLA2) IF ILL LOCAL LBL
2300          2410  25213357335       SP0   SP3   IOR       PUSH FHB  NEXT          FORM AND PUSH EXT LBL, DONE
2301                                    *
2302                                    *
2303                                    *          SCAL, PCAL
2304                                    *     LABEL ON TOS IF N=0, ELSE AT PL-N; PADD = N = CIR(8:15)
2305                                    *     SCAL PATH:  PCL0 IF N=0, PCL1, PCL2,                    PCL5
2306                                    *     PCAL PATH:  PCL0 IF N=0, PCL1,       PCL3 IF EXTERNAL, PCL5
2307                                    *
2308                                    *     FETCH LABEL, EMPTY TOS, CHECK FOR STACK OVFL, IF PCAL STMK
2309                                    *
2310          2411  36327377417       SCAL        PB    CAD       SP0  SF2           SCAL: SF2, SP0 _ -PB-1
2311          2412  02766002433       PCAL        PADD  JMP  PCL0           ZERO    JMP IF N IS ZERO
2312          2413  02167777360             PL    PADD  SUB       BUS  ROP         READ LABEL AT PL-N
2313          2414  37302211744                   JSB  PSHA SP1       SRNZ          SP1=0: EMPTY TOS REGS
2314          2415  23767777302             Z     SM    SUB            HBF         Z<SM OR WRAP AROUND?
2315          2416  26626142527             OPND  JMP  EX11 RC        F1          RC=LABEL: STOV(EX11) IF YES
2316          2417  37766162441       PCL1:             JMP  PCL2                 F2  JMP IF SCAL
2317          2420  24602362672             STA   JSB  STMK RD        UNC         RD=STA: STMK
2318          2421  31726133077             RC    JMP  PCL3 SP2       NEG         SP2=LABEL: JMP IF EXTERNAL
2319                                    *
2320                                    *     ALL RUN INTERRUPTS, SCAL, PCAL AND I/EXIT GO THROUGH PCL5 (OR 6).
2321                                    *     RC (OR UBUS IF PCL6)=LABEL (DP), RD=RSTA, IF F1 PON, IF F2
2322                                    *     I/EXIT AND RA=RQ RB=RS.  FETCH RETURN INSTR, CHECK BOUNDS ON
2323                                    *     AND SET P, SET STA, IF PON CHECK RESTART ENB, IF I/EXIT SET Q,S.
2324                                    *
2325          2422  31761637777       PCL5        RC    ROMN                037777    MASK LABEL ADDR
2326          2423  36137777316       PCL6  UBUS  PB    ADD       BSP0 RNP          READ INSTR
2327          2424  16764777760             PL    UBUS  UBNT                      CHECK BOUNDS
2328          2425  36764777775             SP0   PB    UBNT
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

 2329           2426  37416777775       SP0        INC        P                      SET P
 2330           2427  30506142755            RD    JMP  PW2  STA        F1         STA_RD; JMP IF PON INT
 2331           2430  33437567777            RA    ADD        Q          F2         SET Q; IF NOT I/EXIT
 2332           2431  21437757777            Q     ADD        0          NEXT            REPLACE Q, DONE
 2333           2432  32477757777            RB    ADD        SM         NEXT       SET S, DONE
 2334                                *
 2335                                *    FETCH LABEL FROM TOS FOR S/PCAL
 2336                                *
 2337           2433  37302201732       PCL4            JSB  PUL1 SP1        SRZ        SP1_0, FILL A TOS REG
 2338           2434  37777657577                       ADD             POP  SRL2       S_S-1, LABEL NOW IN RD
 2339           2435  37762361744                       JSB  PSHA            UNC        EMPTY TOS REGS IF SR WAS >1
 2340           2436  23767137762            Z     SM    CAD                  NEG        Z>=SM+1 AND NO WRAP AROUND?
 2341           2437  30626362417            RD    JMP  PCL1 PC           UNC        YES; FINISH LIKE S/PCAL 0
 2342           2440  30206362527            RD    JMP  EX11 PUSH         UNC        NO; REPLACE LABEL, STOV
 2343                                *
 2344                                *    PUSH RETURN ADDR FOR SCAL
 2345                                *
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
 2346           2441  31646132711       PCL5       RC    JMP  CLA2 RB         NEG        STTV(CLA2) IF EXT LABEL
 2347           2442  20217777435            SP0   P     ADD        PUSH CF2            PUSH RET ADDR=(NIR+1)P-PB-1
 2348           2443  24606362422            RD    STA  JMP  PCL5 RD         UNC        NOW RC=LABEL, RD=STA, F2=0
 2349                                *
 2350                                *
 2351                                *         SXIT
 2352                                *    ENTER WITH SR>=1, PADD = -N = -CIR(8:15)
 2353                                *
 2354           2444  36767517560       SXIT PL    PB    SUB              POP  NCRY       BNDV IF P_<PB
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
 2355           2445  33767507776            UBUS RA    SUB                  CRRY       BNDV IF RET ADDR<PL-PB
 2356           2446  17206363013                 SBUS JMP  BNDV PUSH       UNC        REPL ADDR IF BNDV (CTR=0)
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
 2357           2447  36137607310            RD    PB    ADD        RSP0 RNP  SRZ        FETCH RET INSTR; SR=0?
 2358           2450  02762011744                 PADD JSB  PSHA             NZP0         NO, EMPTY TOS IF N<>0
 2359           2451  23317777764            PADD SM    ADD        SP1                 SP1_NEW SM=SM-N
 2360           2452  30202361751            RD    JSB  BNDC PUSH         UNC        REPL ADDR, CK NEW SM
 2361           2453  37416777575            SP0        INC        P    POP            DELETE ADDR, SET P
 2362           2454  01477757777            SP1   ADD        SM         NEXT       SET SM, DONE
 2363                                *
 2364                                *
 2365                                *         EXIT
 2366                                *    ENTRY AT EXIT, WITH PADD = N = CIR(8:15)
 2367                                *
 2368           2455  37762361744       EXIT            JSB  PSHA            UNC        EMPTY TOS, START EXIT AGAIN
 2369           2456  21117777777       EXIT1  Q     ADD        RSP1 ROS            READ DQ, SP1_Q
 2370           2457  23767117776            UBUS SM    CAD                  NCRY       Q>SM?
 2371           2460  37766212455                       JMP  EXIQ            SRN7       YES; EMPTY TOS, START OVER
 2372           2461  21771777774            Q     ROM              177774
 2373           2462  02647777476            UBUS PADD SUB        RB    SP1            RB_Q-N-4 (RET S), SP1
 2374                                *
 2375                                *    F1=EXIT, NF1=IXIT WHICH USES EXIT FROM EXI1
 2376                                *    SP1=Q, RB=PS, IF IXIT RA=RQ, IF EXIT OPND=DQ
 2377                                *    S_Q, FETCH RSTA,DP,X FROM STMK, DISABLE EXT INTS, CK RS,RQ FOR
 2378                                *    STOV, IF EXIT RA_RQ AND CK RS,RQ FOR STUN, IF USER EXIT CK RSTA.
 2379                                *
 2380           2463  37127157774       EXIT1 SP1        CAD        RSP0 ROS  NF1        READ RET STA
```

```
                ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS
2381             2464   26667777054          SP1  OPND SUB       RA   CLSR          IF EXIT RA_Q=DQ, SR_0
2382             2465   01477777777               SP1  ADD       SM                 SM_Q
2383             2466   32767537762          Z    RB   SUB                   NEG     RS>Z?
2384             2467   33767527762          Z    RA   SUB                   POS     NO; RQ>Z?
2385             2470   37306302527               JMP  EX11 SP1              UNC     YES; SP1_0, STOV
2386             2471   37127377775 EXI2     SP0       CAD       BSP0 ROS          READ DP
2387             2472   26606152476               OPND JMP  EXI3 RD             NF1     RD_RSTA; JMP IF IXIT
2388             2473   22767537772          RB   DB   SUB                   NEG     DB>RS?
2389             2474   22767527773          RA   DB   SUB                   POS     NO, DB>RQ?
2390             2475   30766123120               RD   JMP  STUN               POS     YES, STUN IF GOING TO NPRV
2391             2476   37127377775 EXI3     SP0       CAD       BSP0 ROS          READ RET X
2392             2477   26637777777               OPND ADD       RC                 RC_DP
2393             2500   30537537777               RD   ADD       SP3        NEG     SP3_RSTA
2394             2501   24763122770          RD   STA  XOR  SL1              POS     USER CANNOT EXIT TO PRV
2395             2502   37766203117               JMP  TRP6                 NPRV      OR CHANGE EXT INT BIT
2396             2503   24501737777               STA  ROMN       STA  137777       DISABLE EXT INTS SO THAT
2397                                    *                                             ANY PENDING INT OCCURS
2398                                    *                                             IMMEDIATELY AFTER I/EXIT
2399                                    *    OPND=RX, RC=DP, RD=SP3=RSTA
2400                                    *    IF EXTERNAL I/EXIT SET UP SEG, CHECKING FOR NPRV STA TO PRV SEG,
2401                                    *    ABS AND TRACE.  FINISH AT PCL5 (FETCH INSTR, SET STA,P,Q,S).
2402                                    *
2403             2504   30317774777               RD   ADD  RRZ  SP1                ISOLATE SEG #
2404             2505   16771600100               UBUS ROM                000100    READ CST PTR AT 0 IF
2405             2506   16137770177               UBUS ADD  LRZ  BSP0 ROA           SEG#<192, ELSE AT 1
2406             2507   26557777457               OPND ADD       X    CF1           CF1; RESTORE X
2407             2510   24763374410          RD   STA  XOR  RRZ       SF2          SF2; I/EXIT TO SAME SEG?
2408             2511   16766002423               UBUS JMP  PCL6                ZERO  YES, FNSH LIKE PCAL (RC ON
2409             2512   31762302712               RC   JSR  SSEG                UNC   SET UP SEG          \ UBUS)
2410             2513   30777527777               RD   ADD                      POS   STA OR CST PRV?
2411             2514   37766203117               JMP  TRP6                 NPRV      YES, NPRV CAN'T EXIT TO PRV
2412             2515   35773377311          RC   SP2  IOR                HBF          TRACE OR ABS (NF2 IF ABS)
2413             2516   26606152422               SP3  JMP  PCL5 RD            NF1     RD_RSTA; NO, FNSH LIKE PCAL
2414                                    *
2415                                    *    TRACE, ABS, STTV, CSTV AND STOV SYSH CHECKS (F1=PON).
2416                                    *    T,A   RD=TARGET SEG#, SP1<>0, NF2=ABS; FALL THRGH  SYSH; SEG#<2 OR
2417                                    *            FROM I/EXIT, ENTER AT EXIA FROM PCAL/INT          ABS ON ICS
2418                                    *    STTV  RD=SOURCE SEG#, SP1=0; ENTER AT EXI9                SEG#<2
2419                                    *    CSTV  RD=TARGET SEG#, SP1=0, SP2>=0; ENTER AT EX10        SEG#<2
2420                                    *    STOV  SP1=0; ENTER AT EX11                                ON ICS
2421                                    *
2422             2517   02537767457               PADD ADD       SP3  CF1  UNC     CF1; SP3 _ PARAM = N
2423             2520   30502302672 EXIA          RD   JSB  STMK STA       UNC     SET NEW STA AND
2424                                    *                                             STMK IF T,A FROM PCAL
2425             2521   37731740001 EXI9               ROM       SP2  120001       TRACE LABEL=32,1
2426             2522   30761410376 EX10          RD   ROMN            0376 NZRO    SEG# IN RD
2427             2523   37766302762               JMP  SYSH                 UNC     SYS HALT IF SEG#<2
2428             2524   35766123122               SP2  JMP  CSTV                POS     JMP IF CSTV (SP1=0)
2429             2525   01766003123               SP1  JMP  STTV                ZERO    JMP IF STTV
2430             2526   37346163067               JMP  INT5 CTRL       F2          CTR_0; TRACE IF NOT ABS
2431             2527   37731717401 EX11               ROM       SP2  117401       ABS LABEL=31,1
2432             2530   04361400020               CPX1 ROMN       CTRH 0020 ZERO    CTR_0; SYS HALT IF
2433             2531   37766302762               JMP  SYSH                 UNC       STOV OR ABS ON ICS
2434             2532   01766013067               SP1  JMP  INT5                NZRO    ABS TRAP IF NOT STOV
2435             2533   37731714001               ROM       SP2  114001       STOV LABEL_24,1
```

```
2436          2534  16526303020                UBUS JMP  INTO SP3       UNC       PARAM_LABEL
2437                                     *
2438                                     *
2439                                     *       IXIT
2440                                     *   PADD=CIR(12:15); ALSO ENTRY FOR PCN, LOCK AND UNLK
2441                                     *
2442          2535  02766012616    IXI+       PADD JMP  PCN             NZRO      JMP IF PCN, LOCK OR UNLK
2443                                     *
2444                                     *   IXIT EXECUTED ONLY ON THE ICS, BY INTERRUPT PROCEDURES
2445                                     *   OR THE DISPATCHER.  FOR INTERRUPT PROCEDURES: IF REDISPATCH
2446                                     *   REQUESTED (QI)(0)=1, IF DISPATCHER INTERRUPTED (Q)(0)=1
2447                                     *   AND Q<>QI, IF PSDB (QI-18)<>0.
2448                                     *
2449                                     *   (1)  DISPATCHER LAUNCH OF A PROCESS
2450                                     *   (2)  INTERRUPTED PROCESS, RETURN TO PROCESS
2451                                     *   (2A) INT PROC., REDISPATCH REQUESTED BUT PSDB SO RETURN TO PROC.
2452                                     *   (3)  INTERRUPTED INTERRUPT, RETURN TO INTERRUPT
2453                                     *   (4)  INTERRUPTED DISPATCHER, RETURN TO DISPATCHER
2454                                     *   (4A) INT DISP., REDISPATCH REQUESTED BUT PSDB SO RETURN TO DISP.
2455                                     *   (5)  INTERRUPTED PROCESS, REDISPATCH REQUESTED SO START DISPATCHER
2456                                     *   (6)  INTERRUPTED DISP., REDISPATCH REQUESTED SO RESTART DISPATCHER
2457                                     *
2458                                     *   DETERMINE IXIT TYPE: IF FROM EXTERNAL
2459                                     *   INTERRUPT (SEG>1 AND NOT IN DISPATCHER) SEND RIL.
2460                                     *   STMK ELEMENTS ACCESSED FROM MEMORY EVEN IF Q>SM.
2461                                     *   IXIT PATHS:  (1),(2)        IXI2     IXI4
2462                                     *                (2A)      IXI6 IXI2     IXI4
2463                                     *                (3)                IXI3 IXI4
2464                                     *                (4),(4A)  IXI6     IXI3 IXI4
2465                                     *                (5),(6)   IXI6          IXI4
2466                                     *
2467          2536  21117777777                     Q     ADD       BSP1 ROS            READ DQ, SP1_Q
2468          2537  21331777773                     Q     ROM       SP0  177773         SP0_Q-5
2469          2540  37351600010                           ROM       CTRL 000010         CTR_8
2470          2541  04723407056              UBUS CPX1 AND       SP2  CLSR ZERO      SR_0; IN DISPATCHER?
2471          2542  37766302554                           JMP  IXI2           UNC       YES; LAUNCH PROC,ASSUME PRV
2472          2543  24521600376                     STA   ROMN      SP3  000376         ELSE SP3_0 IF SEG<2; SP2=0
2473          2544  14177777775              SP0   CTRL ADD       BUS  ROS            READ DEVICE# AT Q+3
2474          2545  26606203117                     OPND JMP  TRP6 RD         NPRV      RD_DQ; IXIT IS PRV
2475          2546  25766002552                     SP3  JMP  IXI1           ZERO      JMP IF SEG<2 (NO RIL)
2476          2547  26531302000                     OPND ROMI      SP3  102000         FORM RIL CMD
2477          2550  37762361724                          JSB  IOPA           UNC       SEND TO DEVICE (CTR<>0)
2478          2551  04766063055                     CPX1 JMP  INT4           BITB      JMP IF EXT INT (SP2=0)
2479          2552  30766132601    IXI;             RD   JMP  IXI6           NEG       JMP IF RET TO OR START DISP
2480          2553  16766012572                     UBUS JMP  IXI3           NZRO      JMP IF RET TO INTERRUPT
2481                                     *
2482                                     *   (1),(2),(2A); GO TO PROCESS; SP0=QI-5, CTR=8
2483                                     *   SET S-BNK,Q,DL,Z FROM QI-, CLEAR ICS,DISP FLAGS
2484                                     *
2485          2554  37176777175    IXI> SP0       INC       BUS  ROA            READ STACK DB AT QI-4
2486          2555  37731600030                           ROM       SP2  000030         K FOR CLEARING ICS,DISP
2487          2556  37177777175              SP0          ADD       BUS  ROA            READ STDB-BANK
2488          2557  26637777777                     OPND ADD       RC                    SAVE STDB
2489          2560  37127377175              SP0          CAD       BSP0 ROA            READ DS AT QI-6
2490          2561  26157777617                     OPND ADD       SBR  S              SET S-BANK
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:06 PM

2491            2562  37127377175           SP0        CAD        BSP0 ROA             READ DDL AT QI-7
2492            2563  26777777771           RC   OPND  ADD                             Q_S-2 (SAVE 4W STMK
2493            2564  16431777776                UBUS  ROM        Q    177776            IN CASE TRACE OR ABS)
2494            2565  37167377175           SP0        CAD        BUS  ROA             READ DZ AT QI-8
2495            2566  26717777771           RC   OPND  ADD        DL                   SET DL
2496            2567  35777777037                SP2   ADD             CCPX            CLEAR ICS,DISP FLAGS
2497            2570  21117777777                Q     ADD        BSP1 ROS             READ DQ, SP1_Q
2498            2571  26257767771           RC   OPND  ADD        Z         UNC        SET Z, SKIP NEXT LINE
2499                                  #     IXI2 IS MODIFIED BY A PATCH (AFTER SEC 0-7 TO PROPERLY
2500                                  #     COMPUTE PARITY).  -1 STORED AT QI-13 WHENEVER IXI2 IS ENTERED.
2501                                  #     THE OVERLAY OF 2557 CAN BE DONE IN LINE, AND THE THIRD PARITY
2502                                  #     LINE REMOVED IF SEC 4/5 IS REPLACED.
2503                                  #
2504                                  #     (3),(4),(4A); INTERRUPTED INTERRUPT; RD=DQ, SET OPND=DQ(1:15)
2505                                  #
2506            2572  30177773630     IXI3 RD    RD    ADD  SR1  BUS  OPND            OPND_DQ(1:15)
2507                                  #
2508                                  #     (1),(2),(2A),(3),(4),(4A); SET RETURN Q,S IN RA,RB; SP1=Q, OPND=DQ
2509                                  #
2510            2573  01651777774                SP1   ROM        RB   177774          RB_Q-4 (RS)
2511            2574  26667777774           SP1  OPND  SUB        RA                   RA_Q-DQ (RQ)
2512                                  #
2513                                  #     (1),(2),(2A),(3),(4),(4A),(5),(6); SET DB-BNK,DB FROM (Q+1),(Q+2)
2514                                  #     FINISH LIKE EXIT (SET UP RET SEG,STA,X) EXCEPT NO BOUNDS
2515                                  #     CHECKING ON RETURN Q OR S,  SP1=Q, RA=RQ, RB=RS, NF1
2516                                  #
2517            2575  01176777777     IXI4       SP1   INC        BUS  ROS             READ DB-BANK
2518            2576  16176777777                UBUS  INC        BUS  ROS             READ DB
2519            2577  26157777417                OPND  ADD        SBR  DB              SET DB-BANK
2520            2600  26446362463                OPND  JMP  EXI1  DB        UNC        SET DB, FINISH LIKE EXIT
2521                                  #
2522                                  #     (2A),(4),(4A),(5),(6); REDISPATCH REQUESTED OR RETURN TO DISPATCHER
2523                                  #     SET DISPATCHER FLAG, RQ, RS ASSUMING (5) OR (6).  READ (QI), AND IF
2524                                  #     NECESSARY (QI-18), TO DETERMINE PATHS: IXI2_(2A), IXI3_(4),(4A),
2525                                  #     IXI4_(5),(6) AND Q_SP1_QI, (QI)_0.
2526                                  #
2527            2601  05176774177     IXI4       MOD   INC  RR7  BUS  ROA             READ QI ADDR
2528            2602  37731600022                      ROM        SP2  000022          SP2_18
2529            2603  26177777177                OPND  ADD        BUS  ROA             READ (QI)
2530            2604  16677777137                UBUS  ADD        RA   SDFG          SET DISP, RA_QI (RQ)
2531            2605  16651600002                UBUS  ROM        RB   000002          RB_QI+2 (RS)
2532            2606  26766122572                OPND  JMP  IXI3            POS        JMP IF RET TO DISP
2533            2607  35167777173           RA   SP2   SUB        BUS  ROA             READ (QI-18)
2534            2610  21763017773           RA   Q     XOR                  NZRO       Q=QI?
2535            2611  26766012554                OPND  JMP  IXI2            NZRO       YES; RET TO PROC IF PSDB
2536            2612  26766012572                OPND  JMP  IXI3            NZRO       ELSE RET TO DISP IF PSDB
2537            2613  33117777157                RA    ADD        BSP1 WRA             (RE)START DISPATCHER
2538            2614  37177777437                      ADD        BUS  DATA            SP1_QI, (QI)_0
2539            2615  01426362575                SP1   JMP  IXI4  Q         UNC        Q_QI NOW SINCE DISP SET

2540                                  #
2541                                  #
2542                                  #          PCN/LOCK/UNLK
2543                                  #     PUSH CPU#/LOCK/UNLOCK RESOURCE
2544                                  #     PADD = CIR(12:15); INITIAL ENTRY AT IXIT
2545                                  #
```

```
2546          2616  05355133304   PCN   PADD MOD  CRS  SR1  CTRL HBF  NEG        SF1 IF LOCK OR UNLK;
2547                              *                                                   CTR = 2 OR 3, OR 4 OR 5
2548          2617  37762221737                     JSB  PSHM                SR4    EMPTY ONE TOS REG IF PCN
2549          2620  37766263117                     JMP  TRP6                NPRV   PCN,LOCK,UNLK ARE PRV
2550          2621  14355143317               CTRL  CRS  SR1  CTRL HBF  F1         CTR=1 OR 2; SF1 IF UNLK
2551          2622  16217754777               UBUS  ADD  RRZ  PUSH       NEXT       IF PCN TOS=CPU#1 OR 2, DONE
2552                              *
2553          2623  37177777126   LCK1  X           ADD       BUS  ROSA             READ (X), (X) = -1
2554          2624  14721200007               CTRL  ROMX      SP2  000007           SP2=5 IF CPU#2 ELSE 6
2555          2625  26317447377               OPND  ADD       SP1  LBF  NSME       SP1=(X)
2556          2626  37766162623                     JMP  LCK1            F2       READ (X) AGAIN IF -1
2557          2627  14762407774         SP1   CTRL  CAND                ZERO       IF UNLK INT OTHER MODS
2558          2630  37762142641                     JSB  LCK2            F1        REQUESTING RESOURCE IF ANY
2559          2631  37177547146         X           ADD       BUS  WRA  F1
2560          2632  14173367434         SP1   CTRL  IOR       BUS  DATA UNC        LOCK; (X)=(X) 'OR' CPU#
2561          2633  37177757437               ADD       BUS  DATA NEXT       UNLK; (X)=0, DONE
2562          2634  24777532777               STA   ADD SL1                NEG
2563          2635  37766362762               JMP   SYSH                UNC        SYSH IF EXT INTS DISABLED
2564          2636  01766000564         SP1   JMP   NOP                 ZERO       DONE IF (X)=0
2565          2637  20771777777         P     ROM                 177777         ELSE TRY LOCK AGAIN
2566          2640  37766362764               JMP   PAUS                UNC          AFTER INTERRUPT
2567                              *
2568          2641  35177777037   LCK2        SP2   ADD       BUS  CRL             CRL=OTHER CPU'S MOD#
2569          2642  37177707057               ADD       BUS  CMD  RSB              (5 IF CPU#2 ELSE 6)
2570                              *
2571                              *
2572                              *         DISP/PSDB/PSEB
2573                              *     REQUEST RE-DISPATCH/DISABLE/ENABLE DISPATCHER
2574                              *     INITIAL ENTRY AT XCHD; SP2 = PADD = CIR(12:15)
2575                              *     TO ENTER DISPATCHER, INTERRUPT HANDLER (INT1) USED TO
2576                              *     SET UP ICS THEN IXIT PATH (5) FOLLOWED.
2577                              *
2578          2643  05176774177   DISP        MOD   INC  RR7  BUS  ROA             READ QI ADDR
2579          2644  04341600030               CPX1  ROMN      CTRL 000030           CTR NZRO IF ON ICS
2580                              *                                                   (ON ICS IF IN DISP)
2581          2645  37311777756               ROM             SP1  177756           SP1 = -18
2582          2646  26137777176         UBUS  OPND  ADD       BSP0 ROA              READ (QI-18), SP0=QI-18
2583          2647  35766032655               SP2   JMP  PSDE                ODD        JMP IF PSDB OR PSEB
2584          2650  01167777155         SP0   SP1   SUB       BUS  WRA
2585          2651  37171700000               ROM             BUS  100000           (QI)=100000
2586          2652  26777417737               OPND  ADD                 CCE  NZRO     IF (QI-18)=0 AND NOT ON
2587          2653  14766003024   DSP         CTRL  JMP  INT1                ZERO       ICS THEN START DISPATCHER
2588          2654  37777757717               ADD                       CCG  NEXT   CCE IF START DISP, ELSE CCG
2589                              *
2590          2655  00777773377   PSDE        CIR   ADD  SR1                LBF      SF2 IF PSEB, ELSE PSDB
2591          2656  37177567155         SP0         ADD       BUS  WRA  F2
2592          2657  26176757437               OPND  INC       BUS  DATA NEXT       PSDB; INCR (QI-18), DONE
2593          2660  37167047436         UBUS        CAD       BUS  DATA NSME       PSEB; DECR (QI-18)
2594          2661  01167767175         SP0   SP1   SUB       BUS  ROA  UNC        READ (QI) IF (QI-18)
2595          2662  26777757717               OPND  ADD                 CCG  NEXT     WAS 1 OR 0, ELSE DONE
2596          2663  16766002762               UBUS  JMP  SYSH                ZERO     SYS HALT IF (QI-18) WAS 0
2597          2664  15777457737               CTRH  ADD                 CCE  BIT6   SET CCE
2598          2665  37766362670                     JMP  PSD2                UNC      JMP IF NOT IN DISPATCHER
2599          2666  01167777155         SP0   SP1   SUB       BUS  WRA              CLEAR ANY START DISP.
2600          2667  37177767437               ADD       BUS  DATA UNC              REQUESTS, SET CCG, DONE
```

```
2601          2670  26766132653  PSD2       OPND JMP  DSP2           NEG       TEST START DISP. IF
2602          2671  37777757717               ADD                CCG  NEXT        REQ; ELSE SET CCG, DONE
2603                             *
2604                             *
2605                             *     STACK FOUR WORD MARKER
2606                             *
2607          2672  23176777757  STM4       SM   INC        BUS  WRS
2608          2673  37177777426         X    ADD        BUS  DATA          PUSH X
2609          2674  23471600004              SM   ROM        SM   000004    SM_ADDR DQ
2610          2675  16137777757              UBUS ADD        BSP0 WRS
2611          2676  21167777436         UBUS Q    SUB        BUS  DATA          PUSH DQ
2612          2677  23437777777              SM   ADD        Q               Q_ADDR DQ
2613          2700  37127377755         SP0       CAD        BSP0 WRS
2614          2701  24177777437              STA  ADD        BUS  DATA          PUSH STA
2615          2702  20337777777              P    ADD        SP0
2616          2703  37167377755         SP0       CAD        BUS  WRS
2617          2704  36167307435         SP0  PB   CAD        BUS  DATA RSB       PUSH DP, RETURN
2618                             *
2619                             *     CHECK STT LENGTH FOR LABEL FETCHES AND RETURN LABEL,
2620                             *     SP1=0, SP2=STT#, OPND=(PL); LABEL AT PL-STT# RETURNED IN OPND.
2621                             *     VARIOUS INSTRS ENTER AT CLA2 WITH SP1=0 IF STTV DETECTED.
2622                             *
2623          2705  35167777360  CLA8 PL   SP2  SUB        BUS  ROP           READ LABEL AT PL-STT#
2624          2706  26777774777              OPND ADD  RRZ
2625          2707  35767537776         UBUS SP2  SUB                  NEG       LENGTH>=STT#
2626          2710  37777707777              ADD                       RSB       YES, RETURN
2627          2711  24606302521  CLA9       STA  JMP  EX19 RD          UNC       ELSE RD_STA, STTV(EX19)
2628                             *
2629                             *     SET UP CODE SEGMENTS, CHECKING FOR 0<SEG#<=CSTL.  PB,PL SET,
2630                             *     F2_NOT(ABS).  NSTA ( =STA WITH NEW SEG# AND M IF PRV OR CST M)
2631                             *     RETURNED IN RD, AMRT_LENGTH/4 RETURNED IN SP2.  ENTER WITH
2632                             *     SP1=SEG#, SP0=0 OR 1, OPND=(SP0)=CSTP; RANK1 RSB WITH PB_OPND.
2633                             *
2634          2712  24761777400  SSE8       STA  ROMN           177400
2635          2713  16617777774         SP1  UBUS ADD        RD              RD_STA WITH NEW SEG#
2636          2714  26137777177              OPND ADD        BSP0 ROA        GET CSTL
2637          2715  37736437775         SP0       INC        SP2       ODD   IF SEG#<192, SP2_1
2638          2716  01311777500              SP1  ROM        SP1  177500    ELSE SP2_2, SP1_SEG#-192
2639          2717  16766003122              UBUS JMP  CSTV             ZERO  SEG 0 AND 192 DON'T EXIST
2640          2720  01777772774         SP1  SP1  ADD  SL1                 SEG# # 4
2641          2721  16137777175         SP0  UBUS ADD        BSP0 ROA        READ AMRT-LENGTH/4
2642          2722  26767117774         SP1  OPND CAD                  NCRY  SEG# > CSTL?
2643          2723  37306302522              JMP  EX10 SP1              UNC   YES; SP1_0, CSTV(EX10)
2644          2724  37316777775         SP0       INC        SP1
2645          2725  16136777177              UBUS INC        BSP0 ROA        READ BANK
2646          2726  26731240000              OPND ROMI       SP2  020000    SAVE AMRT-LENGTH/4, SET R
2647          2727  16761607777              UBUS ROMN           007777
2648          2730  16777772776         UBUS UBUS ADD  SL1
2649          2731  37227377776         UBUS      CAD        PL              SAVE LENGTH-1
2650          2732  37176777175         SP0       INC        BUS  ROA        READ ADDR
2651          2733  26157777217              OPND ADD        SBR  PB         SET PB-BANK
2652          2734  35777522437              SP2  ADD  SL1        CF2  POS   CF2
2653          2735  30611300000              RD   ROMI       RD   100000    NSTA PRV IF PRV OR M SET
2654          2736  37167377154         SP1       CAD        BUS  WRA
2655          2737  35177537437              SP2  ADD        BUS  DATA NEG  SET R IN CST ENTRY
```

```
      ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS
2656  2740  26237777400           PL   OPND ADD        PL   SF2            SET PL AND F2 IF NOT ABS
2657  2741  26757707777                OPND ADD        PB        RSB      PB_ADDR, RETURN
2658                              *
2659                              *     POWER FAIL AND POWER ON INTERRUPTS; SAVE FF IF PWF.
2660                              *     ABS=BNK_0, F1_1, READ (ZI+1) AND STORE CPX2 AT ZI+1.
2661                              *     IF PWF RETURN; ELSE STA_100000, EXIT TO WAIT IF HALTED
2662                              *     WHEN PWF OR SET UP PON INTERRUPT IF RUNNING WHEN PWF.
2663                              *
2664  2742  37157777017     PWR                ADD        SBR  ABS            ABS=BANK_0
2665  2743  05776774477                MOD  INC  RRZ             SF1          SF1
2666  2744  16176777177                UBUS INC                  ROA          READ ZI
2667  2745  37731721401                     ROM             SP2  121401       PON INT LABEL=35,1
2668  2746  26176777177                OPND INC                  ROA          READ (ZI+1)
2669  2747  16177777157                UBUS ADD                  WRA          WRITE CPX2 AT ZI+1
2670  2750  06177707437                CPX2 ADD                  DATA RSB     RETURN IF PWF
2671  2751  26766022765                OPND JMP  CPRS                 EVEN    HLT IF HLT WHEN PWF (SR=0)
2672  2752  37511700000                     ROM             STA  100000       CLR ITROC,CC
2673  2753  16777773037                UBUS ADD  SR1             CCPX         SET RUN
2674  2754  35526363033                SP2  JMP  INT2 SP3             UNC     PARAM=LABEL; TRAP
2675                              *
2676                              *     CHECK AUTO RESTART AFTER SETTING UP PON INT PROCEDURE
2677                              *
2678  2755  06761410004     PW2        CPX2 ROMN                 0004 NZRO    RUN IF RESTART ENABLED
2679  2756  37777757777                     ADD                      NEXT      ELSE STOP (PON IS PRV)
2680                              *
2681                              *
2682                              *          HALT, PAUS
2683                              *     SPECIAL CONDITIONS RESULTING IN HALT ENTER AT STOP, WAIT OR SYSH
2684                              *
2685  2757  37766263117     HALT               JMP  TRP6                 NPRV     HALT INSTR IS PRV
2686  2760  20411777777     STOP       P       ROM             P    177777       DECR P
2687  2761  15361120600     WAIT       CTRH    ROMX            CTRH 0600 POS     XOR(7,8) ALWAYS POS
2688  2762  15761200640     SYSH       CTRH    ROMX                 000640       XOR(7,8,10) 10=SYSH
2689  2763  15763366036           UBUS CTRH    XOR  SWAB            CCPX UNC     RESET PAN, SET HLT,OPT SYSH
2690  2764  37766263117     PAUS               JMP  TRP6                 NPRV     PAUS IS PRV
2691  2765  37357607761     CPRS  SR           ADD             CTRL      SRZ      CTR_SR
2692  2766  37762361744                        JSB  PSHA                 UNC      EMPTY TOS REGS
2693  2767  37772377777                        REPC
2694  2770  37770327777                        PNLR                      TEST     DISPLAY REGS UNTIL INT
2695  2771  04366363001                CPX1 JMP  IR   CTRH             UNC      CTR_CPX1(9) FOR HMOD PWF
2696                              *
2697                              *
2698                              *     3L  2774-6 USED BY IXI2 PATCH, INCL THIRD PARITY SEC 4/5
```

```
2699                                &              SECTOR 6
2700                                *
2701                                *          INTERRUPT PROCESSOR
2702                                *
2703                                *     INTERRUPTS DETECTED BY FIRMWARE OPTION TEST ENTER AT IRD OR IR.
2704                                *     HARDWARE DETECTED INTERRUPTS ENTER AT IR (VIA LOC 3).
2705                                *     FIRMWARE DETECTED BOUNDS VIOLATIONS ENTER AT BNDV WITH CTR=0.
2706                                *     INT0 AND INT1 ARE ENTRIES FOR ICS INTERRUPTS; SP2=LABEL, SP3=PARAM,
2707                                *       SP1<>-1 (INT0) OR F1=0 (INT1), F2=CLR CONDITION (CTRL;SWAB,CCPX).
2708                                *       INT1 IS ALSO USED BY DISP/PSEB TO CHANGE TO THE ICS.
2709                                *     INT2 IS THE ENTRY FOR COLD LOAD (VIA CODE SHARED WITH SYSTEM DUMP),
2710                                *       AND POWER ON (VIA PWR ROUTINE).
2711                                *     INT4 IS A SHORTCUT ENTRY FOR EXTERNAL INTERRUPTS PENDING
2712                                *       WHEN A PREVIOUS EXTERNAL INTERRUPT IXITS.
2713                                *     INT5 AND INT7 ARE ENTRIES FOR NON-ICS INTERRUPTS; SP2=LABEL,
2714                                *       SP3=PARAM, F1=0, F2=CLR COND; INT7 USES PSHA AND STMK THEN INT5.
2715                                *     TRP6,STUN,STOV,DSTV,CSTV,STTV,TRP7,TRP5,4,3,2,1 AND 0
2716                                *       ARE ENTRIES FOR SPECIFIC NON-ICS INTERRUPTS.
2717                                *     THE HALT MODE INTERRUPTS ARE ENTERED VIA HMOD VIA IR (VIA LOC 3).
2718                                *       SYSH IS THE ENTRY FOR SYSTEM HALTS (DETECTED BY MICRO-CODE).
2719                                *     PWR IS USED BY PWF AND PON INTERRUPTS (PON ENTERS VIA LOC 1).
2720                                *
2721                                *     ENTRY FOR HW DETECTED INTERRUPTS AND INTERRUPTS DETECTED BY TEST.
2722                                *     DETERMINE INTERRUPT TYPE; P_P-1 IF ENTERED AT IRD.
2723                                *     EXIT TO HMOD IF HALTED, WITH SP2_CPX2;
2724                                *     ELSE EXIT TO STOP IF RUN/HALT SW OR NO RUN MODE INTERRUPTS;
2725                                *     ELSE EXIT TO INT1, INT7 OR TRIE WITH APPROPRIATE CONDITIONS,
2726                                *       AND (ZI+1)_CPX2 IF PWF PENDING.
2727                                *
2728                                &3000
2729  3000  20411777777  IRD       P     ROM        P     177777          DECREMENT P
2730                                *
2731  3001  06726023143  IR        CPX2  JMP  HMOD  SP2        EVEN       HMOD IF RUN FF OFF,SP2_CPX2
2732  3002  16766142760            UBUS  JMP  STOP              NEG        STOP IF RUN SW
2733  3003  04521777700            CPX1  ROMN       SP3   177700          SP3_CPX1(0:9)
2734  3004  16761010200            UBUS  ROMX             0200  NZRO       EXTERNAL INTERRUPT ONLY?
2735  3005  37726363024                  JMP  INT1  SP2        UNC          JMP, SP2_0 IF YES
2736  3006  25366002760            SP3   JMP  STOP  CTRH       ZERO       CTR_CPX1(9); STOP IF NO RUN
2737  3007  16766141251            UBUS  JMP  TRIE              NEG        INT; ELSE JMP IF INTEG OVFL
2738  3010  14362032742            CTRL  JSR  PWR   CTRH       ODD        STORE CPX2 AT ZI+1 IF PWF
2739  3011  25772132777            SP3   REPC SL1               NEG        CTR=0 TO 14,
2740  3012  16777532277            UBUS  ADD  SL1         INCT  NEG          FOR CPX1(1:15) ON
2741  3013  14311417771  BNDV      CTRL  ROM        SP1   7771  NZRO       SP1_CTR-7; EXT INT?
2742  3014  37726363024                  JMP  INT1  SP2        UNC          JMP, SP2_0 IF YES
2743  3015  01771600610            SP1   ROM              000610          FORM LABEL
2744  3016  16737776277            UBUS  ADD  SWAB  SP2   INCT            LABEL=CTR+1,1; ADJUST CTR
2745  3017  16537777417            UBUS  ADD        SP3   SF2             PARAM=LABEL; F2=CLEAR INT
2746                                *
2747                                *     ENTRY FOR SOME ICS AND NON-ICS INTERRUPTS;
2748                                *     ALSO USED BY INTERNAL ICS INTERRUPTS ENTERING ABOVE.
2749                                *       ICS INT    ; SP1>= -4, SP2=LABEL, SP3=PARAM OR SP3_MOD#
2750                                *                    IF SP1= -1, F2=CLR COND (IN CTR);
2751                                *       NON-ICS INT; SP1< -4, SP2=LABEL, SP3=PARAM, F2=CLR COND,
2752                                *     F1_0, EXIT TO INT7 IF NON-ICS INTERRUPT.
```

```
                ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:07 PM

2753                                     *
2754                3020  01776417457  INT4      SP1  INC                 CF1  NZRO      CF1: STT=7? (MODULE INT)
2755                3021  05537710777               MOD  ADD  LRZ  SP3                  YES, PARAM=MOD#
2756                3022  01771520004               SP1  ROM                 0004 POS      USE ICS IF STT>=4
2757                3023  37766363140                    JMP  INT7                UNC      ELSE USE USERS STACK
2758                                     *
2759                                     *    ENTRY FOR DISP/PSEB AND SOME ICS INTERRUPTS:
2760                                     *      ALSO USED BY ALL OTHER ICS INTERRUPTS EXCEPT CLD LOAD AND PON.
2761                                     *      INTERNAL INT: SP2=LABEL, SP3=PARAM, F1=0, F2=CLR COND (IN CTR);
2762                                     *      EXTERNAL INT: SP2=0;
2763                                     *      DISP/PSEB   : SP2>0.
2764                                     *    EMPTY TOS, STMK AND DB-BNK,DB; SP0_S, RA_DISP FLAG.
2765                                     *
2766                3024  37762211744  INT5           JSB  PSHA                SRN7      EMPTY TOS REGS
2767                3025  37762362672               JSB  STMK                UNC       STMK
2768                3026  23136777757          SM   INC       BSP0 WRS                 STACK DB-BANK, DATA
2769                3027  04661600010          CPX1 ROMN      RA   000010              (RA NZRO IF IN DISP)
2770                3030  03177777437          RBR  ADD       BUS  DATA                IS IN DB SET OF MCU OPS
2771                3031  37136777755     SP0       INC       BSP0 WRS                 SP0_S
2772                3032  22177777437          DB   ADD       BUS  DATA                STACK DB
2773                                     *
2774                                     *    ENTRY FOR CLD LOAD AND PON:
2775                                     *      ALSO USED BY OTHER ICS INTERRUPTS AND DISP/PSEB FROM INT1.
2776                                     *      CLD LD, PON: SP2=LBL, SP3=PARAM, F1=PON, F2=0, ICS,DISP FLGS CLR:
2777                                     *      OTHER INTERRUPTS AND DISP/PSEB: SP0=S, RA=DISP FLAG.
2778                                     *    S-BNK_0: IF IN DISPATCHER (Q)(0)_1 AND CLR DISP FLAG,
2779                                     *      ELSE CHANGE TO ICS IF NOT ON ICS (Q_QI, Z_ZI, DL_ -1,
2780                                     *      SR_0, (QI-6)_DS, SET ICS FLAG).
2781                                     *
2782                3033  37157777617  INT6           ADD       SBR  S                   S-BANK_0
2783                3034  05116777777               MOD  INC  RRZ  RSP1 ROS               READ QI, SP1_S OR 9
2784                3035  04761410020               CPX1 ROMN           0020 NZRO        (ICS,DISP CLR IF CLD, PON)
2785                3036  37766363045                    JMP  INT3                UNC      JMP IF NOT ON ICS
2786                3037  21177777777               Q    ADD       BUS  ROS               ON ICS: READ DQ
2787                3040  33766003055               RA   JMP  INT4                ZERO     JMP IF NOT IN DISP
2788                3041  16777777037               UBUS ADD                  CCPX         CLEAR DISP FLAG
2789                3042  21177777757               Q    ADD       BUS  WRS               SET DQ(0)=1
2790                3043  26171300000               OPND ROMI      BUS  100000            RETURN TO MEM
2791                3044  37766363055                    JMP  INT4                UNC
2792                3045  01176777777  INT7       SP1  INC       BUS  ROS               NOT ON ICS: READ ZI
2793                3046  26437777117               OPND ADD       Q    SIFG              Q_QI, SET ICS
2794                3047  16311777773               UBUS ROM       SP1  177773            SP1_QI-5
2795                3050  16176777777               UBUS INC       BUS  ROS               READ STDB AT QI-4
2796                3051  26257777057               OPND ADD       7    CLSR              Z_ZI: CLSR IN CASE CLD
2797                3052  37707377777                    CAD       DL                     DL _ -1
2798                3053  37167377754          SP1       CAD       BUS  WRS               QI-6_DS
2799                3054  26167777435          SP0  OPND SUB       BUS  DATA              (MEANINGLESS IF CLD,PON)
2800                                     *
2801                                     *    ENTRY FOR EXT INTERRUPTS DETECTED BY IXIT:
2802                                     *      ALSO USED BY OTHER ICS INTERRUPTS AND DISP/PSEB FROM INT2.
2803                                     *      EXTERNAL INTERRUPTS : SP2=0;
2804                                     *      OTHER ICS INTERRUPTS: SP2=LABEL;
2805                                     *      DISP/PSEB           : SP2>0.
2806                                     *    S_Q+2: IF EXT INTERRUPT SP2_LABEL, SP3_DEV#, STA_140000,
2807                                     *      DB-BNK_0, DB_DBI, F1_0, F2_1, CTR_8.
```

```
      ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:07 PM

2808                                #
2809          3055  21471600002     INT4       Q     ROM                SM    000002          SM_Q+2
2810          3056  35766013067                SP2   JMP   INT5                 NZRO        JMP IF NOT EXT INTERRUPT
2811          3057  11537774457                IOA   ADD   RRZ   SP3   CF1                 SP3_DEV#=PARAM, CF1
2812          3060  16136772776          UBUS  UBUS  INC   SL1   BSP0  ROS                 READ DBI
2813          3061  37511740000                      ROM                STA   140000        CLR STA; SET M,I
2814          3062  37167377775          SP0         CAD                BUS   ROS           READ LABEL
2815          3063  26457777417                OPND  ADD                DB    SF2           DB_DBI, SF2
2816          3064  37157777417                      ADD                SBR   DB            DB=BANK_0
2817          3065  37351600010                      ROM                CTRL  000010        CTR_8
2818          3066  26726353071                OPND  JMP   INT6  SP2                UNC     SP2_LABEL
2819                              #
2820                              #     ENTRY FOR SOME NON-ICS INTERRUPTS;
2821                              #       ALSO USED BY ALL ICS INTERRUPTS AND DISP/PSEB FROM INT4
2822                              #       AND ALL OTHER NON-ICS INTERRUPTS FROM INT7.
2823                              #       INTERRUPTS: SP2=LABEL, SP3=PARAM, F1=PON, F2=CLR COND (IN CTR);
2824                              #       DISP/PSEB : SP2>0.
2825                              #     IF NOT EXT INTERRUPT STA_100000 AND EXIT TO IXI6 IF DISP/PSEB;
2826                              #       PUSH SP3=PARAM INTO MEM, X_CIR, ABS=BNK_0, CLR COND IF F2.
2827                              #
2828          3067  37511700000     INT5             ROM                STA   100000        CLR STA; SET M
2829          3070  35766122601                SP2   JMP   IXI6                 POS         JMP IF DISP/PSEB
2830          3071  23176777757     INT6       SM    INC                BUS   WRS           TOS_PARAM
2831          3072  25177777437                SP3   ADD                BUS   DATA            (IN MEM FOR IXIT
2832          3073  23476777777                SM    INC                SM                     RIL IF EXT INT)
2833          3074  00557777777                CIR   ADD                X                   X_CIR
2834          3075  37157577017                      ADD                SBR   ABS   NF2       ABS_0
2835          3076  14777776037                CTRL  ADD   SWAB               CCPX          CLEAR INT IF F2
2836                              #
2837                              #     SET UP NEW SEG FOR INT PROC AND PCAL; SP2=LABEL, F1=PON, F2=UNKN.
2838                              #     SET P8,PL FROM CST; IF TRACE OR ABS STA_NSTA, STMK, SYSH IF NSEG<2,
2839                              #       IF ABS SYSH IF ICS ELSE ABS TRAP, ELSE TRACE TRAP; CK STT#, CK
2840                              #       TARGET LABEL LOCAL AND IF NPRV CALLABLE; IF STT#=0 TARG LABEL_0.
2841                              #       EXIT TO PCL5, WITH RC_TARGET LABEL, RD_NSTA, F1=UNCH, F2_0.
2842                              #
2843          3077  35317774777     PCL4       SP2   ADD   RR7   SP1                       SP1_SEG# FROM LABEL
2844          3100  16771600100                UBUS  ROM                000100              READ CST PTR AT 0 IF
2845          3101  16137770177                UBUS  ADD   LR7   BSP0  ROA                   SEG#<192, ELSE AT 1
2846          3102  35621677400                SP2   ROMN               RC    077400        ISOLATE STT#
2847          3103  35522352712                SP2   JSB   SSEG  SP3          UNC         SP3_LABEL; SET UP SEG
2848          3104  31637570777                RC    ADD   LR7   RC           NF2         RC_STT#
2849          3105  37177777360          PL          ADD                BUS   ROP           READ STTL IF NOT ABS
2850          3106  35761710000                SP2   ROMN               110000              ISOLATE T,A BITS
2851          3107  16306012520                UBUS  JMP   EXI8  SP1          NZRO        JMP IF T,A SET, ELSE SP1_0
2852          3110  31722352705                RC    JSB   CLA8  SP2          UNC         SP2_STT#, CK STTV (2C JMP)
2853          3111  37731720401                      ROM                SP2   120401        UNCALLABLE, LABEL=33,1
2854          3112  26766132711                OPND  JMP   CLA2               NEG         STTV(CLA2) IF LABEL NOT LOC
2855          3113  16777522437                UBUS  ADD   SL1          CF2   POS         CF2; JMP IF UNCALLABLE,
2856          3114  37766253067                      JMP   INT5                 NPRV          AND NOT COME FROM PRV
2857          3115  31766002422                RC    JMP   PCL5                 ZERO        IF STT=0, LABEL=0; FINISH
2858          3116  26626352422                OPND  JMP   PCL5 RC              UNC           LIKE LOCAL CALL (2C JMP)
2859                              #
2860                              #
2861                              #     ENTRIES FOR NON-ICS INTERRUPTS NOT ENTERING DIRECTLY AT INT5.
2862                              #     TRP6,STUN: SP2_LABEL, SP3_PARAM=LABEL, F1_0, F2_0, EXIT TO INT7.
```

```
2863                                 *    DSTV,CSTV,STTV: SP1=0; FUNCTION SAME AS ABOVE.
2864                                 *    TRP7: SP1=0, F1=0; FUNCTION SAME AS ABOVE.
2865                                 *    TRP5,TRP4: SOV,NEXT IF TRAPS DISABLED
2866                                 *      ELSE CLO, SP2_LABEL, SP3_PARAM, F1_0, CTR_0, EXIT TO INT7.
2867                                 *    TRP3,2,1,0: SP3=0; FUNCTION SAME AS ABOVE.
2868                                 *    INT7: SP2=LABEL, SP3=PARAM, F1=0, F2=CLR COND (IN CTR);
2869                                 *      EMPTY TOS REGS, STMK, EXIT TO INT5.
2870                                 *
2871          3117  37316767777    TRP6          INC       SP1       UNC       5, MODE VIOLATION
2872          3120  37316767777    STUN          INC       SP1       UNC       4, STACK UNDERFLOW
2873          3121  01316762777    DSTV    SP1   INC  SL1  SP1       UNC       3, DSTV
2874          3122  01316762457    CSTV    SP1   INC  SL1  SP1  CF1  UNC       2, CSTV; CF1
2875          3123  01316777457    STTV    SP1   INC       SP1  CF1            1, STTV; CF1
2876          3124  37771600620    TRP           ROM            000620         0, UNIMPL INSTR (FLAGS CLR)
2877          3125  16737776434         SP1  UBUS ADD  SWAB SP2  CF2            LBL=16-21,1; NO INT TO CLR
2878          3126  16526363140         UBUS JMP  INT7 SP3       UNC       USE USERS STK; PARAM=LABEL
2879                                 *
2880          3127  37536767777    TRP3          INC       SP3       UNC       5, FP DIV BY 0
2881          3130  37536767777    TRP4          INC       SP3       UNC       4, INTEGER DIV BY 0
2882          3131  25536762777    TRP3    SP3   INC  SL1  SP3       UNC       3, FP UNDERFLOW
2883          3132  25536762777    TRP2    SP3   INC  SL1  SP3       UNC       2, FP OVFL
2884          3133  25536777777    TRP1    SP3   INC       SP3                 1, INTEGER OVFL
2885          3134  37731714401    TRP0          ROM            114401         0, UNUSED
2886          3135  24777772457         STA  ADD  SL1            CF1            LBL=25,1; CF1, CLO, CTR_0,
2887          3136  16357531636    UBUS UBUS ADD  LLZ  CTRL CLO  NEG       USERS STK IF TRAPS ENABLED
2888          3137  37777757617              ADD                 SOV  NEXT      ELSE SOV, NEXT
2889                                 *
2890          3140  37762211744    INT7          JSB  PSHA            SRNZ      EMPTY TOS REGS
2891          3141  37762362672                  JSB  STMK            UNC       STMK
2892          3142  37766363067                  JMP  INT5            UNC       USERS STK, CLR INT IF F2
2893                                 *
2894                                 *
2895                                 *    SCAN FOR HALT MODE INTERRUPTS.
2896                                 *    ENTER WITH SP2=CPX2, CTRH=CPX1; SP3,F2 MODIFIED.
2897                                 *
2898          3143  14762032742    HMOD       CTRL JSB  PWR            ODD       IF HMOD PWF STORE CPX2 AT
2899          3144  14766033144               CTRL JMP  *              ODD         ZI+1 AND WAIT HERE
2900          3145  35761700000               SP2  ROMN                160000    ISOLATE CPX2(0:2)
*** WARNING (12) *** ZERO,NZRO,NSME SKIP TESTS MADE ON T-BUS
2901          3146  16535002417               UBUS CRS  SL1  SP3  SF2  ZERO      SF2 IN CASE DUMP
2902          3147  37771640000                    ROM                040000    SET RUN IF ANY CPX2(0:2),
2903          3150  16776777037               UBUS INC                CCPX      RESET PANEL FF
2904          3151  25766033202               SP3  JMP  SING           ODD       RUN IF CPX2(0)
2905          3152  16766133221               UBUS JMP  DUMP           NEG       DUMP IF CPX2(1)
2906          3153  35537773437               SP2  ADD  SR1  SP3  CF2            CF2 IN CASE COLD LOAD
2907          3154  25766013206               SP3  JMP  COLD           NZRO      COLD LOAD IF CPX2(2)
2908          3155  35761610000               SP2  ROMN                010000
2909          3156  16766013166               UBUS JMP  LREG           NZRO      LOAD REG IF CPX2(3)
2910          3157  35761604000               SP2  ROMN                004000

2911          3160  16766013167               UBUS JMP  LADR           NZRO      LOAD ADDR IF CPX2(4)
2912          3161  25766053171               SP3  JMP  LMEM           BIT6      LOAD MEM IF CPX2(5)
2913          3162  35766053172               SP2  JMP  DMEM           BIT6      DISPL MEM IF CPX2(6)
2914          3163  25766063202               SP3  JMP  SING           BIT8      EX SING INSTR IF CPX2(7)
2915          3164  35766063203               SP2  JMP  EXSW           BIT9      EX SWCH IF CPX2(8)
2916          3165  37766362760                    JMP  STOP           UNC       DEFAULT IS STOP (NORMAL RET
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2107 PM

2917                                    *                                                 FOR SING,EXSW,HMOD PWF)
2918                                    *
2919            3166  07770767777       LREG      SWCH PNLS                    UNC      LOAD REG
2920            3167  37330377777       LADR           PNLR      SP0                    LOAD ADDR
2921            3170  37766302761                      JMP  WAIT                UNC
2922                                    *
2923            3171  37177767155       LMEM SP0       ADD       BUS  WRA      UNC      LOAD MEM(SP0) WITH SWCH
2924            3172  37177767175       DME. SP0       ADD       BUS  ROA      UNC      DISPLAY MEM(SP0)
2925            3173  07177767437                 SWCH ADD       BUS  DATA     UNC      (SP0)_SWCH
2926            3174  26317777777                 OPND ADD       SP1                    SP1_DISPLAY
2927            3175  35527062777                 SP2  CAD  SL1  SP3           BITB
2928            3176  37336777775            SP0       INC       SP0                    INC ADDR
2929            3177  25777402777                 SP3  ADD  SL1                BITB
2930            3200  37327377775            SP0       CAD       SP0                    DECR ADDR
2931            3201  37766302761                      JMP  WAIT                UNC
2932                                    *
2933            3202  20177767317       SING      P    ADD       BUS  RNP      UNC      EX SINGLE INSTR OR RUN
2934            3203  07177767237       EXSW      SWCH ADD       BUS  NIR      UNC      EX SWCH
2935            3204  20416777777                 P    INC       P                      INC P IF SING OR RUN
2936            3205  37777757777                      ADD                      NEXT
2937                                    *
2938                                    *         COLD LOAD AND DUMP
2939                                    *     F2 IF DUMP; COLD LOAD SHARES FIRST PART OF DUMP ROUTINE,
2940                                    *     READING FROM INSTEAD OF WRITING TO THE SELECTED I/O DEVICE,
2941                                    *     AND IS THEN PROCESSED AS AN INTERNAL "COLD LOAD" INTERRUPT.
2942                                    *
2943                                    *     COLD LOAD ENTERS AT COLD FROM HMOD TO FILL MEM WITH HLT 10S IF
2944                                    *       NOT SWCH(8), OR AT MZR1 (WITH ICS,DISP,SR CLR) BY LOADING RAR
2945                                    *       TO FILL MEM WITH (X); DUMP ENTERS AT DUMP FROM HMOD.
2946                                    *     S-BNK_0, CTR_S-BNK+1, SP1=DEV#, ABS-BNK;SP0;X POSSIBLY MODIFIED.
2947                                    *       IF DEV#<=2 EXIT TO PANEL DIAGNOSTICS; ELSE SP2_OLD LD LABEL, EXIT
2948                                    *       TO DIRECT LD/DMP AT DCLD IF SWCH(9)=1, ELSE SP1_DEV#*4, SR_0.
2949                                    *
2950            3206  07766003221       COLD      SWCH JMP  DUMP                BITB     NO MEM INIT IF SWCH(8)
2951            3207  37551630370       MZR0           ROM            X     030370     X _ HALT 10
2952            3210  37157777017       MZR1           ADD       SBR  ABS             ABS-BANK_0
2953            3211  37317777437                      ADD       SP1  CF2             CF2; SP1_0
2954            3212  37331641374                      ROM       SP0  041374          LH_RUN,ILL ADDR K; RH_ -4
2955            3213  01116777157       MZR2      SP1  INC       BSP1 WRA
2956            3214  37177777426            X         ADD       BUS  DATA            (ABS,SP1)_X
2957            3215  01766013213                 SP1  JMP  MZR2                NZR0     FILL BANK
2958            3216  03156774015            SP0  RBR  INC  RRZ  SBR  ABS             INCR BANK
2959            3217  16766013213                 UBUS JMP  MZR2                NZR0     FILL 4 BANKS
2960            3220  37777771035            SP0       ADD  LLZ       CCPX             CLR ILL ADDR, SET RUN MODE
***  WARNING ( 2)  ***  RBR CONFLICTS WITH PREFETCH ON INSTR ENTRY
2961            3221  03356777617       DUMP      RBR  INC       CTRL S               SAVE S-BANK+1 (CTR NZRO)
2962            3222  37157777617                      ADD       SBR  S               S-BNK_0
2963            3223  07301600077                 SWCH ROMN      SP1  000077          SP1_DEV#
2964            3224  16771527775                 UBUS ROM            7775 POS
2965            3225  37766363520                      JMP  PADT                UNC      JMP TO PAN DIAGS IF DEV#<=2
2966            3226  37731722001                      ROM       SP2  122001          LABEL FOR COLD LOAD
2967            3227  07761400100                 SWCH ROMN           0100 ZERO
2968            3230  37766367773                      JMP  DCLD                UNC      DIRECT LD/DMP IF SWCH(9)
2969            3231  01317772054            SP1  SP1  ADD  SL1  SP1  CLSR            CLSR; SP1_DEV# * 4
2970                                    *
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:07 PM

2971                                    *    LOAD TOS REGS, SP3, SP2, NIR AND OPND WITH (0,SP1)... (SP2 UNCH
2972                                    *    AND NO FETCHES IF NF2=CLD), ABS=BNK_0, SP0_SP1+4, SP1 MODIFIED.
2973                                    *
2974          3232  37157577017     PULA           ADD       SBR  ABS  NF2        ABS=BNK_0
2975          3233  01137767777           SP1  ADD       BSP0 ROS  UNC        READ D0-D4 IF DUMP
2976          3234  01326223244           SP1  JMP  DMP1 SP0       SR4        SKIP REST IF CLD AND SR4
2977          3235  01316637777           SP1  INC       SP1       SRN4       INCR SP1
2978          3236  26537767777           OPND ADD       SP3       UNC        SP3_D4
2979          3237  26206363232           OPND JMP  PUL8 PUSH      UNC        TOS_D0-D3
2980          3240  01177777777           SP1  ADD       BUS  ROS             READ D5
2981          3241  16116777717           UBUS INC       BSP1 RNS             READ D6 INTO NIR
2982          3242  26737777777           OPND ADD       SP2                  SP2_D5
2983          3243  01176777777           SP1  INC       BUS  ROS             READ D7
2984                                    *
2985                                    *    CREATE SECOND PART OF I/O PROGRAM, BEGINNING AT
2986                                    *      ABS=BNK,SP0-1=DEV#*4+3: CONTROL, READ CLD LD PRG (32W INTO D7)
2987                                    *      OR DUMP REC 0 (4K WORDS FROM D6), END WITH INTERRUPT.
2988                                    *    EXIT WITH RH SP0_0 AND IF COLD LOAD STA_0; SP1 MODIFIED.
2989                                    *
2990          3244  37107167155     DMP7 SP0       CAD       BSP1 WRA  F2         SP1_DEV#*4+3
2991          3245  37517777777                  ADD       STA                  CLR I BIT IF NF2=CLD LD
2992          3246  37171640000                  ROM       BUS  040000          IOCW (CONTROL)
2993          3247  37177777155           SP0    ADD       BUS  WRA
2994          3250  07177770437                 SWCH ADD  LRZ  BUS  DATA        IOAW (CONTROL)
2995          3251  37136567155           SP0    INC       BSP0 WRA  F2
2996          3252  37771617740                  ROM            017740
2997          3253  16171600000                 UBUS ROM       BUS  060000      IOCW (R/W)
2998          3254  37136567155           SP0    INC       BSP0 WRA  F2
2999          3255  16776777777                 UBUS INC
3000          3256  16177777437                 UBUS ADD       BUS  DATA        IOAW (D6 OR D7 ADDR)
3001          3257  37176777155           SP0    INC       BUS  WRA
3002          3260  37131634000                  ROM       BSP0 034000          IOCW (END WITH INT)
3003                                    *
3004                                    *    CREATE FIRST PART OF I/O PROGRAM, BEGINNING AT ABS=BNK,DEV#*4:
3005                                    *      I/O PTR, SET BNK=RH SP0.  SEND RIL TO ALL INTERRUPTING DEVICES,
3006                                    *      WAITING FOR CORRECT DEVICE; CHECK I/O PTR IF DUMPING REC 1-N.
3007                                    *    EXIT WITH SP0_SP3, SP1_DEV#*4+7, F1=0; STA PRESERVED; SP3 MODIFIED.
3008                                    *
3009          3261  07301600077     DMP6       SWCH ROMN      SP1  000077      SP1_DEV#
3010          3262  16136772156           UBUS UBUS INC  SL1  BSP0 WRA
3011          3263  37177774435           SP0        ADD  RRZ  BUS  DATA        IOAW (BANK=SP0(14:15))
3012          3264  37127377155           SP0        CAD       BSP0 WRA
3013          3265  37171614000                   ROM       BUS  014000          IOCW (SET BANK)
3014          3266  37167377155           SP0        CAD       BUS  WRA         AT DEV# * 4,
3015          3267  16116777437                 UBUS INC       BSP1 DATA          PUT DEV# * 4 + 1
3016          3270  01531701000                 SP1  ROM       SP3  101000      FORM SIO CMD
3017          3271  25337777777                 SP3  ADD       SP0              SAVE SP3
3018          3272  37762361724     DMP5             JSB  IOPA            UNC   SEND CMD TO DEVICE
3019          3273  37511644000                   ROM       STA  044000          SET I BIT, AND K
3020                                    *                                        FOR CLEARING EXT INT
3021          3274  24537777777                 STA  ADD       SP3              SAVE STA
3022          3275  04777467777                 CPX1 ADD                 BIT8
3023          3276  37766363275                   JMP  *-1                 UNC   WAIT FOR INT
3024          3277  25517777777                 SP3  ADD       STA              REPLACE STA
3025          3300  24777627037                 STA  ADD            CCPX SR4    CLEAR EXT INTERRUPT
```

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:07 PM

3026      3301  37167377174      SP1        CAD        BUS  ROA             READ I/O PTR IF DMP 1-N
3027      3302  11531302000           IOA   ROMI       SP3  102000          FORM RIL CMD
3028      3303  07763377456      UBUS SWCH XOR             CF1             CF1
3029      3304  16761400077           UBUS ROMN                 0077 ZERO
3030      3305  37766363272                JMP  DMP9              UNC       JMP IF NOT CORRECT DEVICE
3031      3306  37762361724                JSB  IOPA              UNC       SEND RIL TO (CORR) DEVICE
3032      3307  01311600006           SP1  ROM        SP1  000006          SP1_DEV##4+7
3033      3310  26767537776      UBUS OPND SUB                    NEG       JMP IF ABN END AND DMP 1-N
3034      3311  37766233345                JMP  DMP5              SRN4        (FOLLOWING SBUS=0)
3035                           *
3036                           *    EXIT TO CLD LD WITH SP3_0 IF NF2;
3037                           *    ELSE SP3_SM, EXIT TO DMP4 IF REC 1-N (SRN4).
3038                           *
3039      3312  37526173033                JMP  INT2 SP3          NF2       JMP IF COLD LOAD
3040      3313  23526233341           SM   JMP  DMP4 SP3          SRN4      JMP IF DMP REC 1-N
3041                           *
3042                           *    PUSH REGS,ETC. INCL D0-5,7 IN TOS,SP0,SP2,OPND, ORG S-BNK(+1)
3043                           *      IN CTR AND ORG SM IN SP3 EXCEPT CPX1,CPX2,D6 AND SIZE INTO
3044                           *      MEM BEGINNING AT S-BNK,SP1+1=DEV##4+8.
3045                           *    PAN (MEM) DIAGS USE DMP3 AS A SUBR TO FIND MEM SIZE; ENTERING WITH
3046                           *      S-BNK=0, SR=0; EXIT (FROM DMP4) WITH SP0_RD+S-BNK=LAST ADDR+1,
3047                           *      (ILL ADDR INT CLEARED), SR_2, SP2,RA-C,CIR MODIFIED.
3048                           *    EXIT WITH SP2_4K, RD_0, RB_CPX1, SR_2, SM_DEV##4+28.
3049                           *
3050      3314  01462211744           SP1  JSB  PSHA SM         SRN7      DUMP D0-D3 (2C JMP)
3051      3315  37217777775      SP0       ADD       PUSH
3052      3316  35217777777           SP2  ADD       PUSH
3053      3317  26202361744           OPND JSB  PSHA PUSH       UNC       DUMP D4,D5,D7
3054      3320  37217777766      X         ADD       PUSH
3055      3321  34217777777           DL   ADD       PUSH
3056      3322  03217777417           RBR  ADD       PUSH DB
3057      3323  22202361744           DB   JSB  PSHA PUSH       UNC       DUMP X,DL,DB-BANK,DB
3058      3324  21217777777           Q    ADD       PUSH
3059      3325  25202361744           SP3  JSB  PSHA PUSH       UNC       DUMP Q,SM
3060      3326  14211777777           CTRL ROM       PUSH 177777          (CTR=S-BANK+1)
3061      3327  37217777762      Z         ADD       PUSH
3062      3330  24202361744           STA  JSB  PSHA PUSH       UNC       DUMP S-BANK,Z,STA
3063      3331  03217777217           RBR  ADD       PUSH PB
3064      3332  36202361744           PB   JSB  PSHA PUSH       UNC       DUMP PB-BANK,PB
3065      3333  20217777777           P    ADD       PUSH
3066      3334  37217777760      PL        ADD       PUSH
3067      3335  00202361744           CIR  JSB  PSHA PUSH       UNC       DUMP P,PL,CIR
3068      3336  37731610000 DMP3          ROM       SP2  010000          SP2_4K
3069      3337  37617777217               ADD       RD   INSR          RD_0, SR=1
3070      3340  04657777217               CPX1 ADD  RB   INSR          RB_CPX1, SR=2
3071                           *
3072                           *    FIND MEM SIZE; PUSH CPX1,CPX2,D6 AND SIZE INTO MEM BEGINNING AT
3073                           *      S-BNK,SM+1=DEV##4+29; DUMP REC 1-N (4K EACH, BEG WITH 0-7777).
3074                           *    ENTER WITH RD=0, RB=CPX1, SR=2, SM=DEV##4+28, ABS-BNK=0, S-BNK=0,
3075                           *      SP1=DEV##4+7, SP2=4K, SP3=ORG SM, NIR=D6.
3076                           *    DMP8 USED TO WRITE REC 1-N; WRITE 4K AND WRITE ADDR SET AT
3077                           *      DEV##4+5,6, RH SP0_BNK AND SM_SP3 BEFORE EACH TRANSFER TO DMP8;
3078                           *      IF ABN I/O PROGRAM END DMP8 ENTERS AT DMP5 WITH SBUS=0 (SR=0)
3079                           *      TO TERMINATE DMP WITH ADDR+BNK OF REC BEING WRITTEN IN CIR.
3080                           *    EXIT TO WAIT AFTER LAST RECORD, WITH ENVIRONMENT UNCHANGED EXCEPT
```

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:07 PM

3081                    *       (DEV#*4) THROUGH (DEV#*4+32), CIR=LAST ADDR+1 + S-BNK=BNK.
3082                    *
3083      3341  35617417770 DMP4 RD   SP2  ADD        PD        NZRO      INC MEM ADDR
3084      3342  03156777617       RBR  INC        SBR  S                 INC S-BANK
3085      3343  16137617770       RD   UBUS ADD    BSP0 ROS  SRN7      READ (ADDR+S-BNK)
3086      3344  16177777237       UBUS ADD        BUS  NIR               NIR_ADDR+BNK IF DMP 1-N
3087      3345  17631601200 DMP5  SBUS ROM        RC   001200            RC_ADDR+BNK+1200
3088      3346  37777777025       RBUS ADD        CCPX                   CLR ILLEGAL ADDR, CIR_NIR
3089      3347  04621600000       CPX1 ROMN       RC   020000            TEST FOR ILLEGAL ADDR INT
3090      3350  31761404003       RC   ROMN            4003 ZERO        256K?
3091      3351  31766003360       RC   JMP  DMP6                ZERO      JMP IF NOT ILLEGAL ADDR
3092      3352  06677707777       CPX2 ADD        RA        RSB          RA_CPX2, RET IF SUBR
3093      3353  37626202761             JMP  WAIT RC        SRZ          RC_0; JMP IF ALL DUMPED
3094      3354  37157777617             ADD       SBR  S                 RESET S-BANK
3095      3355  00202301744       CIR  JSB  PSHA PUSH      UNC          RD_RC; DUMP CPX1,CPX2,D6
3096      3356  37677777215       SP0  ADD        RA   INSR              SIZE=ADDR+BANK
3097      3357  37322301744             JSB  PSHA SP0       UNC          CLR SP0; DUMP SIZE
3098      3360  37766213341 DMP6        JMP  DMP4                SRN7      JMP IF FINDING SIZE
3099      3361  37177777235       SP0  ADD        BUS  NIR               NIR_ADDR+BNK
3100      3362  37107377154       SP1  CAD        BSP1 WRA
3101      3363  30177777437       RD   ADD        BUS  DATA              SET WRITE ADDR
3102      3364  37167377154       SP1  CAD        BUS  WRA
3103      3365  37171600000            ROM        BUS  060000            WRITE 4K
3104      3366  25466303261       SP3  JMP  DMP8 SM        UNC          SET I/O PTR,BANK; SIO
```

```
3105                                &              SECTOR 7
3106                                *
3107                                *          OPTIONAL INSTR JMP TABLE
3108                                *
3109                                *          PANEL DIAGNOSTICS: ADDR TEST, REG TEST, I/O TEST
3110                                *
3111                                *          MEMORY ADDR TEST
3112                                *
3113                                *
3114                                *          OPTIONAL INSTR JMP TABLE
3115                                *     INSTRS 020400/1 THROUGH 020576/7 WHICH ARE IN OPTION
3116                                *     GROUPS CONFIGURED AS PRESENT ENTER AT 34,CIR(9:14)
3117                                *     FROM OPTX, WITH SR=4 AND PADD=CIR(8:15).
3118                                *
3119                                *          OPTION GROUP 0  EFP
3120                                *
3121                                &3400
3122          3400  37766367777                 JMP  UNIM          UNC      /20 EADD/ESUB      020400/1
3123          3401  37766367777                 JMP  UNIM          UNC      /20 EFMP/EDIV            2/3
3124          3402  37766367777                 JMP  UNIM          UNC      /20 ENEG/ECMP            4/5
3125          3403  37766367777                 JMP  UNIM          UNC      /20 TRP7                 6/7
3126          3404  37571610023                 ROM       RAR  010023      /30 EADD/ESUB      020410/1
3127          3405  37571610243                 ROM       RAR  010243      /30 EMPY/EDIV            2/3
3128          3406  37571610701                 ROM       RAR  010701      /30 ENEG/ECMP            4/5
3129          3407  37571607777                 ROM       RAR  007777      /30 TRP7                 6/7
3130                                *
3131                                *          OPTION GROUP 1  APL
3132                                *
3133          3410  02561216011                 PADD ROMX      RAR  016011      LDV/STV  %16031/0  020420/1
3134          3411  02561216103                 PADD ROMX      RAR  016103      MWFV/MWTV 16121/0        2/3
3135          3412  02561216233                 PADD ROMX      RAR  016233      MBFV/MBTV 16217/6        4/5
3136          3413  02561216047                 PADD ROMX      RAR  016047      LDVB/STVB 16061/0        6/7
3137          3414  02561216367                 PADD ROMX      RAR  016367      MVW/      16357/6  020430/1
3138          3415  02561216477                 PADD ROMX      RAR  016477        /       16445/4        2/3
3139          3416  02571616737                 PADD ROM       RAR  016737        /       16773/4        4/5
3140          3417  02571616737                 PADD ROM       RAR  016737        /       16775/6        6/7
3141                                *
3142                                *          OPTION GROUP 2
3143                                *
3144          3420  37766366757                      JMP  L200          UNC            %06757   020440/1
3145          3421  37766366761                      JMP  L202          UNC             06761         2/3
3146          3422  37766366763                      JMP  L204          UNC             06763         4/5
3147          3423  37766366765                      JMP  L206          UNC             06765         6/7
3148          3424  37766366767                      JMP  L210          UNC             06767   020450/1
3149          3425  37766366771                      JMP  L212          UNC             06771         2/3
3150          3426  37766366773                      JMP  L214          UNC             06773         4/5
3151          3427  37766366775                      JMP  L216          UNC             06775         6/7
3152                                % L200  6757
3153                                % L202  6761
3154                                % L204  6763
3155                                % L206  6765
3156                                % L210  6767
3157                                % L212  6771
3158                                % L214  6773
```

```
3159                                % LOC  6775
3160                                *
3161                                *       OPTION GROUP 3
3162                                *
3163        3430  02571615677               PADD ROM        RAR  015677          %15757/60 020460/1
3164        3431  02571615677               PADD ROM        RAR  015677           15761/62       2/3
3165        3432  02571615677               PADD ROM        RAR  015677           15763/64       4/5
3166        3433  02571615677               PADD ROM        RAR  015677           15765/66       6/7
3167        3434  02571615677               PADD ROM        RAR  015677           15767/70 020470/1
3168        3435  02571615677               PADD ROM        RAR  015677           15771/72       2/3
3169        3436  02571615677               PADD ROM        RAR  015677           15773/74       4/5
3170        3437  02571615677               PADD ROM        RAR  015677           15775/76       6/7
3171                                *
3172                                *       OPTION GROUP 4
3173                                *
3174        3440  02571617657               PADD ROM        RAR  017657          %17757/60 020500/1
3175        3441  02571617657               PADD ROM        RAR  017657           17761/62       2/3
3176        3442  02571617657               PADD ROM        RAR  017657           17763/64       4/5
3177        3443  02571617657               PADD ROM        RAR  017657           17765/66       6/7
3178        3444  02571617657               PADD ROM        RAR  017657           17767/70 020510/1
3179        3445  02571617657               PADD ROM        RAR  017657           17771/72       2/3
3180        3446  02571617657               PADD ROM        RAR  017657           17773/74       4/5
3181        3447  02571617657               PADD ROM        RAR  017657           17775/76       6/7
3182                                *
3183                                *       OPTION GROUP 5
3184                                *
3185        3450  02571623637               PADD ROM        RAR  023637          %23757/60 020520/1
3186        3451  02571623637               PADD ROM        RAR  023637           23761/62       2/3
3187        3452  02571623637               PADD ROM        RAR  023637           23763/64       4/5
3188        3453  02571623637               PADD ROM        RAR  023637           23765/66       6/7
3189        3454  02571623637               PADD ROM        RAR  023637           23767/70 020530/1
3190        3455  02571623637               PADD ROM        RAR  023637           23771/72       2/3
3191        3456  02571623637               PADD ROM        RAR  023637           23773/74       4/5
3192        3457  02571623637               PADD ROM        RAR  023637           23775/76       6/7
3193                                *
3194                                *       OPTION GROUP 6
3195                                *
3196        3460  02571624617               PADD ROM        RAR  024617          %24757/60 020540/1
3197        3461  02571624617               PADD ROM        RAR  024617           24761/62       2/3
3198        3462  02571624617               PADD ROM        RAR  024617           24763/64       4/5
3199        3463  02571624617               PADD ROM        RAR  024617           24765/66       6/7
3200        3464  02571624617               PADD ROM        RAR  024617           24767/70 020550/1
3201        3465  02571624617               PADD ROM        RAR  024617           24771/72       2/3
3202        3466  02571624617               PADD ROM        RAR  024617           24773/74       4/5
3203        3467  02571624617               PADD ROM        RAR  024617           24775/76       6/7
3204                                *
3205                                *       OPTION GROUP 7 (ALWAYS PRESENT)
3206                                *
3207        3470  37766347233                   JMP  DMUL             F3       NOT DMUL/DDIV      020560/1

3208        3471  37777777777                   ADD                                                     2/3
3209        3472  37777777777                   ADD                                                     4/5
3210        3473  37777767777                   ADD                   UNC      TRP7               6/7
3211        3474  37766367233                   JMP  DMUL             UNC      DMUL/DDIV          020570/1
3212        3475  37777777777                   ADD                                                     2/3
3213        3476  37777777777                   ADD                                                     4/5
```

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:07 PM

3214  3477  37777777777          ADD                                                    4/7
3215  3500  37766367777          JMP  UNIM            UNC       OPTION GROUP 7 TO TRP7
3216                      *
3217                      *
3218                      *          PANEL DIAGNOSTICS
3219                      *     ENTER AT PADT FROM COLD LOAD (OR DUMP) IF DEV#<3
3220                      *     (AFTER MZRO IF SWCH(8)), WITH S-BANK=0, SP1=DEV#.
3221                      *
3222                      *          PANEL MEMORY TEST
3223                      *     EXIT TO PANEL REG TEST IF DEV#<>0.
3224                      *     TEST (0,0) THROUGH (LAST LEGAL ADDR) USING MEM OR N##2 TESTS,
3225                      *     N##2 IF SWCH(0), (SEE MEM AND N##2 TEST FUNCTIONAL DESCRIPTIONS).
3226                      *
3227                      &35>0
3228  3520  01766013527  PADT      SP1  JMP  PRGT            NZRO      JMP IF NOT MEMORY TEST
3229  3521  37442363336               JSB  DMP3 DB            UNC       DB_0; FIND LAST ADDR+1
3230  3522  37247047050         RD    CAD       Z     CLSR NSMF      SR_0;
3231  3523  37147377615         SP0   CAD       SBR   S                S=BNK,Z_ENDING ADDR
3232  3524  07157531417         SWCH  ADD  LLZ  SBR   DB   NEG       DB=BNK,DB_STARTING ADDR
3233  3525  37706363777               JMP  ATST DL            UNC       DL_0; MEM TEST IF NO
3234  3526  37706367775               JMP  NQT1 DL            UNC         SWCH(0), ELSE N##2
3235                      *
3236                      *
3237                      *          PANEL REG TEST
3238                      *     EXIT TO PANEL I/O TEST IF DEV#<>1.
3239                      *     TEST REGS, NOP AND UBUS; BASE(CIR)+1, BASE+2... STORED INTO
3240                      *       REG1, REG2... THEN REGS ARE CHECKED, THEN BASE IS INCREMENTED
3241                      *       AND THE TEST REPEATS UNTIL TERMINATED BY RUN/HALT SW (RESULTING
3242                      *       IN AN EXIT TO WAIT WITH A VALID ENVIRONMENT AFTER CURRENT PASS)
3243                      *       OR AN ERROR IS DETECTED.
3244                      *     IF ERROR PAUSE (IN RUN MODE) WITH CIR=BAD BITS (REG XOR
3245                      *       CORRECT DATA) UNTIL RUN/HALT SW (TEST). THEN EXIT TO WAIT
3246                      *       WITH CIR=REG# (SEE REG STORE LINES BELOW).
3247                      *     SP1, SP2, SP3, STA, CIR, CTR, NOP ON RBUS AND SBUS
3248                      *       AND UBUS ON SBUS MUST WORK TO SOME EXTENT TO REACH THE
3249                      *       PANEL REG TEST AND DETECT AND DISPLAY ANY ERRORS.
3250                      *
3251  3527  01766023702  PRGT      SP1  JMP  PIOT            EVEN      JMP IF NOT REG TEST
3252  3530  37311600230  PRG1           ROM       SP1   000230         SP1_NIRTOCIR,CLR ICS,DISP K
3253  3531  00176777237              CIR  INC       BUS   NIR
3254  3532  00316777777              CIR  INC       SP1                SP1_BASE+1
3255  3533  01777777037              SP1  ADD            CCPX         BASE_BASE+1; CLR ICS,DISP
3256  3534  37157777417                   ADD       SBR   DB           SET UP VALID STACK
3257  3535  37157777617                   ADD       SBR   S              AND CODE SEG SETW
3258  3536  37711602000                   ROM       DL    002000         0,002000 AND 0,007777
3259  3537  37451603000                   ROM       DB    003000         WITH ICS,DISP FLAGS CLR.
3260  3540  37431604000                   ROM       Q     004000         LOW CORE AND ICS
3261  3541  16477777057              UBUS ADD       SM    CLSR           NOT INITIALIZED.
3262  3542  37251604777                   ROM       Z     004777
3263  3543  37157777217                   ADD       SBR   PB
3264  3544  37751606000                   ROM       PB    006000
3265  3545  37411607000                   ROM       P     007000
3266  3546  37231607777                   ROM       PL    007777
3267  3547  37511700003                   ROM       STA   100003           STA_100003
3268  3550  06766132761              CPX2 JMP  WAIT            NEG       TERMINATE IF RUN/HALT SW
```

```
3269                                    *
3270          3551  01236777117         SP1   INC         PL    SIFG        PL - BASE + 1; SET ICS
3271          3552  16256777777         UBUS  INC         Z                 Z      "      2
3272          3553  16556777777         UBUS  INC         X                 X      "      3
3273          3554  16616777777         UBUS  INC         RD                RD     "      4
3274          3555  16636777777         UBUS  INC         RC                RC     "      5
3275          3556  16656777777         UBUS  INC         RB                RB     "      6
3276          3557  16676777777         UBUS  INC         RA                RA     "      7
3277          3560  16336777777         UBUS  INC         SP0               SP0    "     10
3278          3561  16356777777         UBUS  INC         CTRL              CTRL   "     11
3279          3562  16416777777         UBUS  INC         P                 P      "     12
3280          3563  16436777777         UBUS  INC         Q                 Q      "     13
3281          3564  16456777777         UBUS  INC         DB                DB     "     14
3282          3565  16476777777         UBUS  INC         SM                SM     "     15
3283          3566  16516777777         UBUS  INC         STA               STA    "     16
3284          3567  16536777777         UBUS  INC         SP3               SP3    "     17
3285          3570  16176777637         UBUS  INC         BUS   OPND        OPND   "     20
3286          3571  16716777777         UBUS  INC         DL                DL     "     21
3287          3572  16736777777         UBUS  INC         SP2               SP2    "     22
3288          3573  16756777777         UBUS  INC         PB                PB     "     23
3289          3574  16016777777         UBUS  INC         PCLK              PCLK   "     24
3290                                    *
3291          3575  37773377774   SP1         IOR                           CK SP1 GATING, SBUS NOP,
3292          3576  16762363672           UBUS  JSB   PRGS          UNC       RBUS AND SBUS UBUS
3293          3577  37777777760   PL          ADD
3294          3600  16762363672           UBUS  JSB   PRGS          UNC     CK PL, RBUS NOP
3295          3601  37777777762   Z           ADD
3296          3602  16762363672           UBUS  JSB   PRGS          UNC     CK Z
3297          3603  37777777766   X           ADD
3298          3604  16762363672           UBUS  JSB   PRGS          UNC     CK X
3299          3605  37777777770   RD          ADD
3300          3606  16762363672           UBUS  JSB   PRGS          UNC     CK RBUS RD
3301          3607  37777777771   RC          ADD
3302          3610  16762363672           UBUS  JSB   PRGS          UNC     CK RBUS RC
3303          3611  37777777772   RB          ADD
3304          3612  16762363672           UBUS  JSB   PRGS          UNC     CK RBUS RB
3305          3613  37777777773   RA          ADD
3306          3614  16762363672           UBUS  JSB   PRGS          UNC     CK RBUS RA
3307          3615  37777777775   SP0         ADD
3308          3616  16762363672           UBUS  JSB   PRGS          UNC     CK SP0
3309          3617  01761777700           SP1   ROMN          177700
3310          3620  14777777776   UBUS  CTRL  ADD
3311          3621  16762363672           UBUS  JSB   PRGS          UNC     CK CTRL
3312          3622  20762363672           P     JSB   PRGS          UNC     CK P
3313          3623  21762363672           Q     JSB   PRGS          UNC     CK Q
3314          3624  22762363672           DB    JSB   PRGS          UNC     CK DB
3315          3625  23762363672           SM    JSB   PRGS          UNC     CK SM
3316          3626  24762363672           STA   JSB   PRGS          UNC     CK STA
3317          3627  25762363672           SP3   JSB   PRGS          UNC     CK SP3
3318          3630  26762363672           OPND  JSB   PRGS          UNC     CK OPND
3319          3631  34762363672           DL    JSB   PRGS          UNC     CK DL
3320          3632  35762363672           SP2   JSB   PRGS          UNC     CK SP2
3321          3633  36762363672           PB    JSB   PRGS          UNC     CK PB
3322          3634  13762363672           PCLK  JSB   PRGS          UNC     CK PCLK
3323                                    *
```

```
3324          3635  01617777777          SP1  ADD        RD                  RD - BASE + 25
3325          3636  16636777777          UBUS INC        RC                  RC    "     26
3326          3637  16656777777          UBUS INC        RB                  RB    "     27
3327          3640  16676777777          UBUS INC        RA                  RA    "     30
3328          3641  16376777777          UBUS INC        CTRH                CTRH  "     31
3329          3642  16156777017          UBUS INC        SBR  ABS            A-BNK "     32
3330          3643  16156777217          UBUS INC        SBR  PB             P-BNK "     33
3331          3644  16156777417          UBUS INC        SBR  DB             D-BNK "     34
3332          3645  16156777617          UBUS INC        SBR  S              S-BNK "     35
3333                              #
3334          3646  30762363672          RD   JSB  PRG5             UNC       CK SBUS RD
3335          3647  31762363672          RC   JSB  PRG5             UNC       CK SBUS RC
3336          3650  32762363672          RB   JSB  PRG5             UNC       CK SBUS RB
3337          3651  33762363672          RA   JSB  PRG5             UNC       CK SBUS RA
3338          3652  01761770077          SP1  ROMN            170077
3339          3653  15777777776     UBUS CTRH ADD
3340          3654  16762363672          UBUS JSB  PRG5             UNC       CK CTRH
3341          3655  01761777774          SP1  ROMN            177774
3342          3656  03777777016     UBUS RBR  ADD             ABS
3343          3657  16762363672          UBUS JSB  PRG5             UNC       CK ABS-BNK
3344          3660  01761777774          SP1  ROMN            177774
3345          3661  03777777216     UBUS RBR  ADD             PB
3346          3662  16762363672          UBUS JSB  PRG5             UNC       CK PB-BNK
3347          3663  01761777774          SP1  ROMN            177774
3348          3664  03777777416     UBUS RBR  ADD             DB
3349          3665  16762363672          UBUS JSB  PRG5             UNC       CK DB-BNK
3350          3666  01761777774          SP1  ROMN            177774
3351          3667  03777777616     UBUS RBR  ADD             S
3352          3670  16762363672          UBUS JSB  PRG5             UNC       CK S-BNK
3353          3671  37766363530               JMP  PRG1             UNC       NEXT PASS
3354                              #
3355                              #    COMPARE REG ON UBUS WITH SP1.  IF NO ERROR SP1_SP1+1, RETURN;
3356                              #    ELSE PAUSE (IN RUN MODE) WITH CIR=UBUS XOR SP1 UNTIL RUN/HALT
3357                              #    SWITCH (TEST,STA=%200), THEN EXIT TO WAIT WITH CIR=ERROR #.
3358                              #
3359          3672  01163017236     PRG= UBUS SP1  XOR        BUS  NIR  NZRO      NIR_BAD BITS
3360          3673  01316707777          SP1  INC        SP1       RSB       IF NO ERR INCR SP1, RETURN
3361          3674  37511600200               ROM        STA  000200         ELSE STA_NIR TO CIR K
3362          3675  00167777234          SP1  CIR  SUB        BUS  NIR          NIR_ERROR #
3363          3676  24772377037               STA  REPC             CCPX      CIR_BAD BITS
3364          3677  37770327777                    PNLR                  TEST  PAUSE (SP1=CORRECT DATA)
3365          3700  24777777037               STA  ADD              CCPX      CIR_ERROR #
3366          3701  37766362761                    JMP  WAIT             UNC       EXIT TO WAIT
3367                              #
3368                              #
3369                              #         PANEL I/O TEST
3370                              #    SP1=2.  SEND TIO TO DEVICES 3 THROUGH %177; PAUSING (IN RUN MODE)
3371                              #    WITH CIR=DEV# OF FIRST RESPONDING DEV UNTIL RUN/HALT SW, THEN
3372                              #    WAIT (IN HALT MODE) WITH CIR=DEV STA UNTIL RN/HLT (TEST,STA<%200),
3373                              #    THEN TEST NEXT DEV; AFTER LAST DEV EXIT TO WAIT WITH CIR=%200.
3374                              #
3375          3702  01116777237     PIO= SP1  INC        BSP1 NIR          NIR_SP1_DEV#+1
3376          3703  37631640201               ROM        RC   040201       RC_RUN,CIR_NIR,PAN FF K
3377          3704  16777777037          UBUS ADD             CCPX      SET RUN,CIR_DEV#,RESET PAN
3378          3705  01506062761          SP1  JMP  WAIT STA        BIT8      DONE IF STA_DEV#=%200
```

```
3379          3706  01031702400            SPI  ROM       IOA  102400        SEND TIO CMD TO DEV
3380          3707  04777433777            CPX1 ADD  SR1            ODD       I/O TIMEOUT?
3381          3710  12722303762            IOD  JSB  ATS6 SP2       UNC       NO; DISPL DEV# THEN DEV STA
3382          3711  37766303702                 JMP  PIOT            UNC       TEST NEXT DEVICE
3383                                *
3384                                *
3385                                *        MEMORY TEST
3386                                *    MEMORY TEST (ENTRY AT ATST) BEGINS WITH ONE OF THE
3387                                *    FOLLOWING OPERATIONS AND EXECUTES THEM IN THE SEQUENCE SHOWN.
3388                                *      F1 SP3(15) OPERATION
3389                                *       0      0   FILL MEMORY LOCATIONS UPWARD WITH THEIR OWN
3390                                *                    ADDR+BANK; READ BACK UPWARD.
3391                                *       1      0   FILL MEMORY LOCATIONS UPWARD WITH PARITY (0 OR -1)
3392                                *                    CALCULATED ON THEIR ADDRS; READ BACK UPWARD.
3393                                *       0      1   FILL MEMORY LOCATIONS DOWNWARD WITH THE COMPLEMENT
3394                                *                    OF THEIR ADDR+BANK; READ BACK UPWARD.
3395                                *       1      1   FILL MEMORY LOCATIONS UPWARD WITH COMPL(PARITY)
3396                                *                    CALCULATED ON THEIR ADDRS; READ BACK UPWARD.
3397                                *    MEMORY IS TESTED FROM DB-BANK,DB THROUGH S-BANK,Z.
3398                                *    TEST REPEATS UNTIL TERMINATED BY RUN/HALT SW (EXIT TO WAIT AFTER
3399                                *    CURRENT OPERATION) OR AN ERROR IS DETECTED (SEE READ BACK BELOW).
3400                                *    F3 SHOULD BE ON UNLESS AN ERROR IS DETECTED, AND NAMER IS
3401                                *    DECREMENTED BEFORE EACH OPERATION.
3402                                *
3403                                *    PARITY CALCULATION: RA=ADDR, CTR(S)=PARITY OF PREVIOUS ADDR;
3404                                *    RETURN WITH F2_CTR(S)_PARITY OF ADDR.  THE STARTING ADDR, IN DB,
3405                                *    DETERMINES WHETHER PARITY OR COMPL(PARITY) IS EVEN OR ODD.
3406                                *
3407          3712  37767377773   ATP1 RA        CAD                          CALC PREVIOUS ADDR
3408          3713  16772027277            UBUS REPC           INCT EVEN     COMPL P FOR THE 1 ADDED
3409          3714  16777423277            UBUS ADD  SR1       INCT EVEN       AND FOR EACH 1 DELETED
3410          3715  14777707377            CTRL ADD            LBF  RSR        WHEN ADDR IS INCREMENTED
3411                                *
3412                                *    INITIALIZATION: MEMORY TEST ENTRY POINT AT ATST.
3413                                *    F1 AND POSSIBLY SP3(15) ARE COMPLEMENTED TO GET INITIAL OPERATION.
3414                                *
3415          3716  06766132761   ATS1       CPX2 JMP  WAIT              NEG     TERMINATE IF RUN/HALT SW
3416          3717  22217557477              DB   ADD       PUSH SF1    NF1     RA_DB, SR_NZRO: COMPL F1
3417          3720  25536777457              SP3  INC       SP3  CF1            COMPL SP3(15) IF NF1
3418          3721  16357427377              UBUS ADD       CTRL LBF  EVEN     F2_CTR(S)_SP3(15)
3419          3722  03737557617              RBR  ADD       SP2  S    NF1      INIT ADDR=S-BNK,Z
3420          3723  03737767417              RBR  ADD       SP2  DB   UNC        IF NF1 AND F2,
3421          3724  37677777762         Z         ADD       RA                     ELSE INIT ADDR=DB-BNK,DB
3422          3725  35157777017              SP2  ADD       SBR  ABS          ABS=BANK_INIT BANK
3423          3726  37631643201                   ROM       RC   043201       RC_RUN,DPE,CIR_NIR,PAR EF K
3424                                *
3425                                *    FILL MEMORY: ABS=BNK,RA=ADDR; NF1 AND F2 IF DOWNWARD.
3426                                *
*** WARNING ( 2) *** RBR CONFLICTS WITH PREFETCH ON INSTR ENTRY
3427          3727  03657777017   ATS2       RBR  ADD       RB   ABS          RB_ABS=BNK; SF3
3428          3730  16317557073         RA   UBUS ADD       SP1  SF3  NF1    IF NF1 SP1_ADDR+BNK
3429          3731  37302303712              JSB  AT10 SPI       UNC        ELSE SP1_0, CALC PARITY
3430          3732  33177567157              RA   ADD       BUS  WRA  F2     STORE SP1 IF UPWARD
3431          3733  01177767437              SP1  ADD       BUS  DATA UNC      ADDR TEST OR PARITY=0
3432          3734  01167157437              SP1  CAD       BUS  DATA NF1    ELSE STORE COMPL SP1
```

```
     ADDRESS   CONTENTS      LABEL RBUS SBUS FUNC SHFT STOR SPEC SKIP       COMMENTS                FRI, AUG 13, 1976,  2:07 PM

3433   3735  37766363771                    JMP  ATS8                UNC      FILLING UPWARD IF F1 OR NF2
3434   3736  22723017773          RA   DB   XOR       SP2            NZRO     ELSE DOWN; CK BNK,ADDR
3435   3737  03723377412          RB   RBR  XOR       SP2  DB                  SP2_0 IF LAST BNK,ADDR,
3436   3740  37667047773          RA        CAD       RA             NSME       ADDR_ADDR-1, DECR BNK IF
3437   3741  16157777012          RB   UBUS ADD       SBR  ABS                 ADDR= -1 (NO AFFECT IF 0)
3438   3742  35766013727    ATS3       SP2  JMP  ATS2                NZRO     FILL NEXT ADDR IF NOT DONE
3439                         *
3440                         *     READ BACK; IF ERROR (BAD DATA OR ILL ADDR,CPU TIMER,PE) CF3 AND
3441                         *       DSPL (IN RN MODE) BAD BITS (CORRECT XOR ACTUAL DATA) IN CIR UNTIL
3442                         *       RN/HLT SW, THEN WAIT (IN HLT MODE) WITH CIR=ADDR+CPX1(2:6) (ALSO
3443                         *       ADDR IN ABS-BNK,SP0, CORRECT DATA IN SP1, ACTUAL DATA IN OPND)
3444                         *       UNTIL RUN/HALT SW (TEST,STA=0), THEN CONTINUE WITH NEXT ADDR.
3445                         *     N**2 TEST USES ATS5 TO DISPLAY BAD DATA AND ADDR;
3446                         *       STA=0, RC=040201, NIR=BAD BITS, RB=CPX1 BITS, SP2,RA=ADDR.
3447                         *     PANEL I/O TEST USES ATS6 TO DISPLAY DEV#S AND DEV STA;
3448                         *       STA<%200, RC=040201, NIR=DEV#, SP2=DEV STA.
3449                         *
3450   3743  25357777057               SP3  ADD       CTRL CLSR           SR_0; INIT CTR FOR PARITY
3451   3744  22677777777               DB   ADD       RA                  INITIALIZE ADDR; RA_DB
3452   3745  03777777417               RBR  ADD            DB             INITIALIZE BANK;
3453   3746  16157777017               UBUS ADD       SBR  ABS              ABS-BNK_DB-BNK
3454   3747  33137777177    ATS4       RA   ADD       BSP0 ROA            READ (ABS-BNK,RA) UPWARD
***  WARNING ( 1)  ***  RBR MAY CONFLICT WITH PREVIOUS BANK SELECTION
3455   3750  03317557013          RA   RBR  ADD       SP1  ABS  NF1       IF NF1 SP1_ADDR+BNK
3456   3751  37302363712               JSB  AT10 SP1            UNC      ELSE SP1_0, CALC PARITY
3457   3752  03657577017               RBR  ADD       RB   ABS  NF2       RB_ABS-BNK; COMPL SP1 IF
3458   3753  01307377777               SP1  CAD       SP1                   DWN ADDR TEST OR PARITY=1
3459   3754  04721637000               CPX1 ROMN      SP2  037000          SP2_NZRO IF ILL ADDR,TM,PE
3460   3755  26163017234          SP1  OPND XOR       BUS  NIR  NZRO      NIR_BAD BITS IF ANY
3461   3756  35766003766               SP2  JMP  ATS7                ZERO     JMP IF NO ERROR
3462                         *
3463   3757  35737776772    ATS5 RB    SP2  ADD  SWAB SP2                 SP2_ADDR(0:3),BNK,
3464   3760  33761770001               RA   ROMN           170001            CPX1(2:6),ADDR(15)
3465   3761  35737777776          UBUS SP2  ADD       SP2
3466                         *
3467   3762  31777777037    ATS6       RC   ADD            CCPX           RUN,DPE,CIR_NIR,PAN
3468   3763  35172377237               SP2  REPC      BUS  NIR            NIR_ADDR,ETC. (SP2)
3469   3764  06737531177               CPX2 ADD  LLZ  SP2  CF3  NEG       CF3; DISPL BAD BITS
3470   3765  35773377031          RC   SP2  IOR            CCPX           TOG MODE TO HLT,CIR_NIR,PAN
3471   3766  33172347237    ATS7       RA   REPC      BUS  NIR  F3        HLT WITH CIR=ADR IN ABS,SP0
3472   3767  37770327777               PNLR                    TEST             CORRECT DATA IN SP1
3473   3770  37767307031          RC   CAD                CCPX RSB             ACTUAL DATA IN OPND
3474                         *                                             RN,DPE,CIR_NIR; RET IF SUBR
3475   3771  33723017062    ATS8 Z     RA   XOR       SP2  SF3  NZRO      SF3; CHECK BNK,ADDR;
3476   3772  03723377612          RB   RBR  XOR       SP2  S               SP2_0 IF LAST BNK,ADDR,
3477   3773  33676417777               RA   INC       RA             NZRO     ADDR_ADDR+1,
3478   3774  32156777017               RB   INC       SBR  ABS              INCR BNK IF ADDR=0
3479   3775  37766213742                    JMP  ATS3                SRNZ     JMP IF FILLING (UPWARD)
3480   3776  35766013747               SP2  JMP  ATS4                NZRO     READ BACK IF NOT DONE
3481   3777  37506363716    ATS9            JMP  ATS1 STA            UNC      ELSE STA_0, NEXT PASS
3482                         #0377 1K PARITY
3483                         #0777 1K PARITY
3484                         #1377 1K PARITY
3485                         #1777 1K PARITY
3486                         #2377 1K PARITY
```

```
3487                                 *2777 1K PARITY
3488                                 *3377 1K PARITY
3489                                 *3501 1K PARITY
3490          0777  32312035756      &0777 2K PARITY
3491          1777  00366610515      &1777 2K PARITY
3492          2777  06563506317      &2777 2K PARITY
3493                                 *3501 2K PARITY
3494                                 *
3495                                 *
3496                                 *    DSG2 PATCH, SEE 2373 IN DSG2; REQUIRES REBURN 4/5 0-7.
3497                                 *    RESTORE DB-BNK FROM CTR IF MDS (AND EITHER DSEG IS ARS).
3498                                 *    SECOND PARITY LINE ADDED TO SECTOR 4/5.
3499                                 *
3500                                 &2373
3501          2373  01526203140              SP1  JMP  INT7 SP3        NPRV     ALL DSEG USERS ARE PRV
3502          2374  00761410020              CIR  ROMN              0020 NZRO     CIR(11)=0 FOR MDS ONLY
3503          2375  14157777417              CTRL ADD       SBR  DB             DB=BNK_CTR IF MDS
3504          2376  37766303140                   JMP  INT7                UNC   TRAP
3505          2377  23150104577      &2377 2K PARITY                            SECOND PARITY LINE
3506                                 *
3507                                 *
3508                                 *    IXI2 PATCH, SEE 2557 IN IXI2; REQUIRES REBURN 4/5 0-7.
3509                                 *    STORE -1 AT SP0-CTR=QI-13 FOR IXIT PATHS (1), (2), (2A).
3510                                 *    THIRD PARITY LINE ADDED TO SECTOR 4/5.
3511                                 *
3512                                 &2557
3513          2557  26622302775              OPND JSB  IX2A RC         UNC      OVERLAY 2557
3514                                 &2775
3515          2775  14167777155    IX2A SP0  CTRL SUB       RUS  WRA           (QI-13)_ -1
3516          2776  37167307437              CAD       RUS  DATA RSB          RETURN
3517          2774  14762712217      &2774 2K PARITY                          THIRD PARITY LINE
3518                                 *
3519                                 *
3520                                 *    SIO PATCH, SEE 1626 IN SIO;
3521                                 *      REQUIRES NEW 2/3 6 AND REBURN 2/3 0-5,7.
3522                                 *    READ DRT PTR AT SP1#4 (AFTER WRITE).
3523                                 *    SECOND PARITY LINE ADDED TO SECTOR 2/3.
3524                                 *
3525                                 &1626
3526          1626  37762301764              JSB  SIO2            UNC      OVERLAY 1626
3527                                 &1764
3528          1764  01177702174    SIO2 SP1  SP1  ADD  SL1  BUS  ROA          READ I/O PTR
3529          1765  37766301724              JMP  IOPA            UNC      IOPA, RET WILL BE TO SIO
3530          1776  01166304167      &1776 2K PARITY                          SECOND PARITY LINE
```

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:07 PM

3531                                &              SECTOR 14
3532                                *
3533                                *          N**2 MEMORY TEST
3534                                *
3535                                *          DMUL/DDIV
3536                                *
3537                                *
3538                                *          N**2 MEMORY TEST
3539                                *     INITIALIZATION: ENTRY POINTS AT NOTS AND NQT1(2).
3540                                *     DB=BNK,DB=FIRST BLOCK, S=BNK,Z=LAST BLOCK.
3541                                *     TEST FOR INTERLEAVED MEMORY, PB_2WI FLAG;
3542                                *     ABS=BNK,SP1 AND RA_STARTING ADDR OF FIRST BLOCK,
3543                                *     RD_DELTA ENDING ADDR IN BLOCK, F2_1.
3544                                *
3545                                &7000
3546      7000  37157777017     NQ01          ADD        SBR   ABS             ABS=BANK_0
3547      7001  37176777157                   INC        BUS   WRA             STORE 1 AT ADDR 1
3548      7002  37176777437                   INC        BUS   DATA
3549      7003  22311200020           DB      ROMI       SP1   000020          SP1_DB(0:3)...
3550      7004  37177777025     RBUS          ADD        BUS   CRL             STORE 0 AT ADDR 1 IN MOD 0;
3551      7005  37176777057                   INC        BUS   CMD               IF INTERLEAVED ADDR 1=1
3552      7006  37177777437                   ADD        BUS   DATA              ELSE ADDR 1=0
3553      7007  37176777177                   INC        BUS   ROA
3554      7010  37611607777                   ROM        RD    007777          RD_BLOCK RANGE (4K-1)
3555      7011  26757407417           OPND    ADD        PB    SF2  ZERO       PB_2WI FLAG (0=F,1=T), SF2
3556      7012  30617772777           RD      ADD  SL1   RD                    BLOCK RANGE_8K-2 IF 2WI
3557      7013  16302771774     SP1   UBUS    CAND LLZ   SP1                   ABS=BNK,SP1 AND RA_STARTING
3558      7014  03677771416     UBUS  RBR     ADD  LLZ   RA    DB                ADDR=DB=BNK,DB(0:2,3),0
3559      7015  17157777017           SBUS    ADD        SBR   ABS
3560                                *
3561                                *     FILL BLOCK WITH 0'S OR 1'S (DL): ABS=BNK,SP1 AND RA=STARTING ADDR,
3562                                *     RD=DELTA ENDING ADDR, PB=2WI FLAG; RC_CMPL MCUDP,CLR DPE K'S,
3563                                *     SP0_FIELD STARTING ADDR, SP3_ENDING ADDR, SR_0, CTR_0, SP1 MODIF.
3564                                *
3565      7016  37631607406     NQ1A          ROM        RC    007406          RC_CMPL MCUDP,CLR DPE K'S
3566      7017  30537777054           SP1     RD   ADD   SP3   CLSR            SP3_ENDING ADDR, SR_0
3567      7020  36116777154     NQ1B  SP1     PB   INC   BSP1  WRA             FILL BLOCK, EXCEPT STARTING
3568      7021  34762771031     NQ1C  RC      DL   CAND LLZ   CCPX              ADDR, WITH 17 BITS OF
3569      7022  34177777437                   DL   ADD   BUS   DATA              0'S (EVEN PARITY) IF DL=0
3570      7023  34762701031           RC      DL   CAND LLZ   CCPX RSB          OR 17 BTS OF 1'S IF DL=-1
3571      7024  25343007774           SP1     SP3  XOR   CTRL  ZERO             RET IF SUBR; CTR_0
3572      7025  33326367020                   RA   JMP   NQ11 SP0   UNC         SP0_FIELD STARTING ADDR
3573                                *
3574                                *     TEST BLOCK: MOVING 1'S (P=1) THROUGH FIELD OF 0'S (P=0).
3575                                *     DL=0, ABS=BNK,SP0 AND RA=STARTING ADDR, SP3=ENDING ADDR, RD=DELTA
3576                                *     ENDING ADDR, PB=2WI FLG, RC=CMPL MCUDP,CLR DPE, F2=1, SR=0, CTR=0.
3577                                *     IF ERROR EXIT TO NQ80-3 WITH ABS=BNK,SP0_FIELD ADDR,
3578                                *     RA_TEST ADDR, SP1_BAD DATA BITS, RB_SELECTED CPX1 BITS, DL=0;
3579                                *     DATA ERRORS ARE DETECTED BEFORE CPX1 ERRORS.
3580                                *     ELSE EXIT TO NQ30 WITH SP0_ENDING ADDR, SP1,RA,RB,F1 MODIFIED.
3581                                *
3582      7026  33177777157     NQ1D          RA   ADD   BUS   WRA
3583      7027  34707777457                   DL   CAD   DL    CF1             CF1; WRITE TEST WORD,
3584      7030  34302367021                   DL   JSB   NQ12  SP1   UNC         PRESERVING DL IN SP1
```

```
3585          7031  03763377013          RA   RBR  XOR                ABS              NIR_TEST ADDR(0:9)+BNK
3586          7032  14163377236          UBUS CTRL XOR        BUS   NIR                (CTR=TEST ADDR(10:15))
3587          7033  37771700103                ROM                  100103
3588          7034  37767375036          UBUS       CAD   RLZ        CCPX             RUN,CLR ILLEGAL ADDR
3589          7035  37777775025          RBUS       ADD   RLZ        CCPX             RUN,CLR CPU TIMER
3590          7036  37776776025          RBUS       INC   SWAB       CCPX             RUN,NIRTOCIR,CLR SPE
3591          7037  37767375031          RC         CAD   RLZ        CCPX             CLR APE
3592          7040  31777775037                RC   ADD   RLZ        CCPX             CLR DPE
3593          7041  01706137062                SP1  JMP   NQ23 DL          NEG        DL_SP1; JMP IF FIELD OF 1'S
3594          7042  37177767175          SP0        ADD         BUS  ROA  UNC        READ FIRST 0'S WORD
3595          7043  36136777175 NQ14     SP0  PB    INC         BSP0 ROA             READ 0'S WORD
3596          7044  04641637000                CPX1 ROMN         RB   037000          RB_CPX1(2:6), JMP
3597          7045  16766017134                UBUS JMP  NQ80                NZRO       IF ERR READING 1'S
3598          7046  25767777315          SP0   SP3  SUB                HBF              SF1 IF NOT END OF BLOCK
3599          7047  33177777177                RA   ADD         BUS  ROA             READ 1'S WORD
3600          7050  26306017126                OPND JMP  NQ83 SP1         NZRO       JMP IF POSS ERR IN 0'S DATA
3601          7051  31777775037                RC   ADD  RLZ        CCPX             ELSE CLR EXPECTED DPE
3602          7052  04641411000                CPX1 ROMN         RB   1000 NZRO       IN 0'S, RB_CPX1(6),
3603          7053  37766367132                     JMP  NQ82                UNC        OTHER ERRS CNT AS 1'S
3604          7054  26307017777 NQ1<           OPND CAD         SP1            NZRO     JMP IF NO ERR IN 1'S
3605          7055  37766147043                     JMP  NQ14                F1         AND NOT END OF BLOCK
3606          7056  01766017133                SP1  JMP  NQ81                NZRO       JMP IF ERR IN 1'S DATA
3607          7057  04641637000                CPX1 ROMN         RB   037000          RB_CPX1(2:6), JMP IF
3608          7060  16766017134                UBUS JMP  NQ80                NZRO       ERR READING 1'S LAST
3609          7061  37766367102                     JMP  NQ30                UNC        END OF BLOCK
3610                                *
3611                                *     TEST BLOCK: MOVING 0'S (P=0) THROUGH FIELD OF 1'S (P=1).
3612                                *     DL= -1, ABS=BNK,SP0 AND RA=STARTING ADDR, SP3=ENDING ADDR, RD_DELTA
3613                                *     ENDING ADDR, PB=2WI, RC=CMPL MCUDP,CLR DPE, F2=1, F1=0, SR,CTR=0.
3614                                *     IF ERROR EXIT TO NQ80-3 WITH ABS=BNK,SP0_FIELD ADDR,
3615                                *     RA_TEST ADDR, SP1_BAD DATA BITS, RB_SELECTED CPX1 BITS, DL= -1;
3616                                *     DATA ERRORS ARE DETECTED BEFORE CPX1 ERRORS.
3617                                *     ELSE SP1,RA,RB,F1 MODIFIED.
3618                                *
3619          7062  37177767175 NQ2<  SP0        ADD         BUS  ROA  UNC        READ FIRST 1'S WORD
3620          7063  36136777175 NQ24   SP0  PB    INC         BSP0 ROA             READ 1'S WORD
3621          7064  31777775037                RC   ADD  RLZ        CCPX             DPE EXPECTED IN 0'S IF NOT
3622          7065  04641411000                CPX1 ROMN         RB   1000 NZRO       STARTING ADDR, RB_CPX1(6)
3623          7066  37766147134                     JMP  NQ80                F1         OTHER ERRS CNT AS 1'S
3624          7067  25767777315          SP0   SP3  SUB                HBF              SF1 IF NOT END OF BLOCK
3625          7070  33177777177                RA   ADD         BUS  ROA             READ 0'S WORD
3626          7071  26307007777                OPND CAD         SP1       ZERO
3627          7072  37766367126                     JMP  NQ83                UNC        JMP IF POSS ERR IN 1'S DATA
3628          7073  04641637000                CPX1 ROMN         RB   037000          RB_CPX1(2:6), JMP
3629          7074  16766017132                UBUS JMP  NQ82                NZRO       IF ERROR READING 1'S
3630          7075  26306017133 NQ2<           OPND JMP  NQ81 SP1         NZRO       JMP IF ERR IN 0'S DATA
3631          7076  37766147063                     JMP  NQ24                F1         JMP IF NOT END OF BLOCK
3632          7077  31777775037                RC   ADD  RLZ        CCPX             RB_CPX1(6), CLR EXPECTED
3633          7100  04641411000                CPX1 ROMN         RB   1000 NZRO       DPE IN READING 0'S LAST,
3634          7101  37766367134                     JMP  NQ80                UNC        OTHER ERRS CNT AS 1'S
3635                                *
3636          7102  33177777157 NQ3<           RA   ADD         BUS  WRA             RESTORE TEST LOC TO
3637          7103  37762367021                     JSB  NQ12                UNC        BACKGROUND FIELD
3638          7104  30327777775          SP0   RD   SUB                SP0              RESET FIELD ADDR
3639          7105  25767777313          RA    SP3  SUB                HBF              SF1 IF NOT DONE WITH BLOCK
```

```
      ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:08 PM

3640          7106  36676557773          RA   PB   INC       RA        NF1       RA,CTR_NEXT TEST ADDR, JMP
3641          7107  16346367026               UBUS JMP  NQ13 CTRL      UNC         IF NOT DONE WITH BLOCK
3642                                    *
3643                                    *    CHECK FOR END OF PASS; S=BNK,Z=LAST BLOCK,
3644                                    *    ABS-BNK,SP3=ENDING ADDR IN CURRENT BLOCK, PB=2WI FLAG; SR_0,
3645                                    *    IF 2WI, TEST EVEN THEN ODD ADDRS OF CURRENT BLOCK BEFORE
3646                                    *    GOING ON TO NEXT BLOCK.  IF PASS NOT COMPLETE EXIT TO NQ10
3647                                    *    WITH ABS-BNK,SP1 AND RA_STARTING ADDR IN BLOCK TO BE TESTED.
3648                                    *    MAIN PANEL INITIAL ENTRY AT NQT5 WITH SR=0;
3649                                    *    FRONT PANEL ENTRY AT NQT1 WITH DL=0, F3=0, SR=0.
3650                                    *
***  WARNING ( 2)  ***  RBR CONFLICTS WITH PREFETCH ON INSTR ENTRY
3651          7110  03677777017  NQ40       RBR  ADD       RA   ABS             RA_ABS-BNK; SP1_FIRST ADDR
3652          7111  25316417057               SP3  INC       SP1  CLSR NZRO       OF NEXT BLOCK IF NOT 2WI
3653          7112  33156777017               RA   INC       SBR  ABS             INCR BANK IF NEC; SR_0
3654          7113  36763427774          SP1  PB   AND                 EVEN      SP1_FIRST ADDR+1 OF CURRENT
3655          7114  30307777774          SP1  RD   SUB       SP1                  BLK IF NOT DONE WITH 2WI
3656          7115  03763017613          RA   RBR  XOR             S    NZRO      CURRENT BLOCK LAST BLOCK?
3657          7116  25767117762          Z    SP3  CAD                  NCRY
3658          7117  01666367016               SP1  JMP  NQ10 RA        UNC       NO; RA_BEG ADDR, TEST NEXT
3659          7120  34707367777               DL   CAD       DL        UNC       YES; CMPL DL (BACKGR FIELD)
3660          7121  16703767077  NQT4         UBUS AND       DL   SF3  UNC       MP INIT ENTRY; DL_0, F3_1
3661          7122  21656767436          UBUS Q    INC       RB   CF2  UNC       INCR PASS IF DL=0 & <> INIT
3662          7123  16643777437  NQT2         UBUS AND       RB   CF2            RB_0 IF INIT PASS; FP ENTRY
3663          7124  16426347145               UBUS JMP  NQ35 Q             F3    PRINT PASS CTR IF MP
3664          7125  37506367000  NQ41              JMP  NQ01 STA         UNC       STA_0, NEXT PASS
3665                                    *
3666                                    *
3667                                    *    ERROR PROCESSING; POSSIBLE ERRORS IN FIELD DATA ENTER AT NQ83,
3668                                    *    OTHERS ENTER AT NQ80-2; DATA ERRORS DETECTED BEFORE CPX1 ERRORS.
3669                                    *    F2=1, ABS-BNK,SP0=FIELD ADDR, RA=TEST ADDR, SP1=BAD DATA BITS,
3670                                    *    RB=SELECTED CPX1 BITS, DL=0 IF 0'S FIELD OR -1 IF 1'S FIELD, SR=0.
3671                                    *    IF ERROR; SR_3 IF ENTRY AT NQ83; FIELD DATA ERROR
3672                                    *              SR_2 IF ENTRY AT NQ82; CPX1 ERROR IN READING FIELD DATA
3673                                    *              SR_1 IF ENTRY AT NQ81; TEST DATA ERROR
3674                                    *              SR=0 IF ENTRY AT NQ80; CPX1 ERROR IN READING TEST DATA
3675                                    *    RET TO NQ15 OR NQ25 IF NO ERROR; ELSE PRINT ERRORS ON DEV# 3
3676                                    *    (CLK/TTY) IF F3 (N##2 INITIATED FROM MP) OR USE FP DISPLAY ROUTINE
3677                                    *    AT AT55 IN MAIN MICRO-CODE IF NF3 (N##2 INITIATED FROM FP).
3678                                    *    RET TO NQ40; P,CIR,SP2,RC,CTR,F1 MAY BE MODIFIED.
3679                                    *    NQ40 ENTERS AT NQ85 WITH F2=0 TO PRINT RB=PASS CTR; RET TO NQ41.
3680                                    *
3681          7126  33403017775  NQ83 SP0  RA   XOR       P         NZRO      ERR IF FIELD<>TEST ADR ELSE
3682          7127  34766007054               DL   JMP  NQ15             ZERO      RET TO 0'S FLD W/O DPE CK
3683          7130  20766007075               P    JMP  NQ25             ZERO      OR TO 1'S FLD W/O CPX1 CK
3684          7131  37777777217                    ADD                INSR      SR_3
3685          7132  37777777217  NQ82             ADD                INSR      SR_2
3686          7133  37777777217  NQ81             ADD                INSR      SR_1
3687          7134  37417777475  NQ80 SP0         ADD       P    SF1           P_FIELD ADDR, SF1
3688          7135  01177777237               SP1  ADD       BUS  NIR           NIR_BAD BITS FOR FP DISPLAY
3689          7136  37631640201                    ROM       RC   040201        RC_K FOR FP DISPLAY
3690          7137  03367347017               RBR  CAD       CTRH ABS  F3      CTR_CTRM; SP2_BNK,
3691          7140  17722143757               SBUS JSB  AT55 SP2         F1        FP DISPLAY IF FP (2C JMP)
3692          7141  16762347167               UBUS JSB  NQ87             F3      PRINT BANK IF MP
3693          7142  16762347157               UBUS JSB  NQ86             F3      PRINT FIELD ADDR IF MP
```

```
                ADDRESS   CONTENTS      LABEL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:08 PM

3694            7143  33402347157             RA   JSB  NQ86 P          F3        PRINT TEST ADDR IF MP
3695            7144  01402347157             SP1  JSB  NQ86 P          F3        PRINT BAD DATA BITS IF MP
3696            7145  32402347157   NQ8⊆      RB   JSB  NQ86 P          F3        PRINT SEL CPX1 BITS IF MP
3697            7146  34761700000             DL   ROMN                100000
3698            7147  16417777761        SR   UBUS ADD       P
3699            7150  37762347157             JSB  NQ86                F3        PRINT DL(0),SR IF MP
3700            7151  37051600015             ROM       IOD  000015
3701            7152  37762347170             JSB  NQ88                F3        CARRIAGE RET IF MP
3702            7153  37051600012             ROM       IOD  000012
3703            7154  37762347170             JSB  NQ88                F3        LINE FEED IF MP
3704            7155  37766167110             JMP  NQ40                F2        NEXT BLOCK IF F2
3705            7156  37766367125             JMP  NQ41                UNC       ELSE NEXT PASS
3706                                  *
3707                                  *
3708                                  *  SUBROUTINE TO OUTPUT 16-BIT NUMBER IN P TO CLK/TTY
3709                                  *  WHICH IS IN DEV#3 AND FOLLOWS WITH A SPACE
3710                                  *
3711                                  *  ENTRY POINT NQ87 CONVERTS 3 BIT NUMBER ON UBUS TO ASCII
3712                                  *  AND PRINTS IT
3713                                  *
3714                                  *  ENTRY POINT NQ88 PRINTS DATA ALREADY IN IOD AND WAITS FOR
3715                                  *  COMPLETION
3716                                  *
3717                                  *  F1=OUTPUT SPACE AFTER NUMBER
3718                                  *  CTR=-6 FOR A 6 DIGIT NUMBER
3719                                  *
3720            7157  20415372477   NQ8⊄      P    CRS  SL1  P    SF1             SF1; P_NUMBER&SL1
3721            7160  37351777772             ROM       CTRL 177772             CTR_-6 (PRINT 6)
3722            7161  20761600001             P    ROMN           000001          LAST BIT ONLY
3723            7162  16766367167             UBUS JMP  NQ87           UNC       GO CONV AND PRINT
3724                                  *
3725            7163  20775372777   NQ8⊇      P    CRS  SL1                       SHIFT IN NEXT DIGIT
3726            7164  16775372777             UBUS CRS  SL1
3727            7165  16415372777             UBUS CRS  SL1  P
3728            7166  16761600007             UBUS ROMN           000007          LAST 3 BITS
3729            7167  16051600060   NQ8⊇      UBUS ROM       IOD  000060        CONVERT TO ASCII
3730            7170  37031701403   NQ8⊇           ROM       IOA  101403        WIO CHARACTER
3731            7171  37031702403   NQ9⊂           ROM       IOA  102403        TIO
3732            7172  12761410400             IOD  ROMN           0400 NZRO       WAIT FOR COMPLETION
3733            7173  37766367171                  JMP  NQ90           UNC
3734            7174  37777737277                  ADD             INCT CTR⊇       DONE?
3735            7175  37766367163                  JMP  NQ89           UNC       NO
3736            7176  37347147457                  CAD         CTRL CF1  F1        CTR_-1, CF1
3737            7177  37777707777                  ADD                   RSB       RET IF DONE
3738            7200  37051600040                  ROM       IOD  000040        SPACE
3739            7201  37766367170                  JMP  NQ88           UNC
3740                                  *
3741                                  *


3742                                  *          DMUL/DDIV
3743                                  *     ENTER VIA JMP TABLE IN SEC 7 WITH SR=4, RD,RC=U, RB,RA=V.
3744                                  *     F1_WSGN, U_ABS(U), V_ABS(V), SP2_ORG MSU; EXIT TO DDIV IF DDIV.
3745                                  *
3746                                  &7233
3747            7233  30722137262   DMUL      RD   JSB  DM1A SP2        NEG       SP2_ORG MSU, COMPL U IF NEG
3748            7234  32762137265             RB   JSB  DM1B             NEG       COMPL V IF NEG; F1=WSGN
```

```
      ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:08 PM

3749          7235  00766037300             CIR  JMP  DDIV           ODD       JMP IF DDIV
3750                                #
3751                                #    DELETE TWO ELEMENTS FROM
3752                                #    THE STACK.  MULTIPLY LSU*LSV LEAVING THE RESULT IN SP3,SP2.
3753                                #    CLEAR OVERFLOW.  SINCE LSU*LSV=LSV*LSU, IT DOES NOT MATTER
3754                                #    IF U AND V ARE LATER SWAPPED.
3755                                #
3756          7236  31537777637           RC   ADD       SP3  CLO            SP3_LSU: CLO
3757          7237  37772607237                REPN           DCSR 20        DCSR
3758          7240  16774333273     RA    UBUS MPAD SR1       INCT CTRM      LSU*LSV
3759          7241  17537777237           SBUS ADD       SP3  DCSR           SP3_MSW OF RESULT:DCSR
3760          7242  25737777777           SP3  ADD       SP2                 SP2_LSW OF RESULT
3761                                #
3762                                #    IN ORDER NOT TO OVERFLOW, EITHER MSU OR MSV MUST EQUAL
3763                                #    ZERO.  IF MSU<>0 THEN U AND V ARE EXCHANGED AND THE NEW
3764                                #    MSU MUST EQUAL ZERO.  IF NOT, AN ADDITIONAL MULTIPLY
3765                                #    MUST BE PERFORMED TO INSURE THE ANSWER IS CORRECT MODULO 2**32.
3766                                #
3767          7243  30766007247     DMU3        RD   JMP  DMU3           ZERO      JMP IF MSU=0
3768          7244  32337417257                 RB   ADD       SP0  INCN NZRO      SP0_MSU
3769          7245  31777767257                      ADD                 INCN UNC    SWAP U AND V IF MSU<>0
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
3770          7246  32522367271                 RB   JSB  DM2A SP3        UNC       OVFL IF MSU,MSV<>0 (1C JMP)
3771                                #
3772                                #    IF MSV=0 THIS MULTIPLY CAN BE SKIPPED SINCE ITS RESULT
3773                                #    WILL BE ZERO.  IF NOT, MULTIPLY MSV*LSU.  OVERFLOW OCCURS
3774                                #    IF THE MSW OF THIS PRODUCT IS NOT ZERO.
3775                                #
3776          7247  32526007254     DMU4        RB   JMP  DMU4 SP3        ZERO      SKIP MPAD IF ZERO (2C+ JMP)
3777          7250  25772607777                 SP3  REPN                  20
3778          7251  16774333271           RC    UBUS MPAD SR1       INCT CTRM      MSV*LSU
3779          7252  17777407777           SBUS ADD                    ZERO
3780          7253  25777777617           SP3  ADD                 SOV           OVF IF NZRO;UBUS=SP3
3781                                #
3782                                #    IF THE ANSWER IS NOT POSITIVE AT THIS POINT, AN OVERFLOW
3783                                #    HAS OCCURRED UNLESS THEN RESULT IS EXACTLY -2**31 AND A
3784                                #    NEGATIVE ANSWER IS EXPECTED.  NEGATE THE RESULT IF F1 IS SET
3785                                #
3786          7254  16642137274     DMU4        UBUS JSB  DM4A RB           NEG       POSSIBLE OVF IF NEG
3787          7255  35666150563                 SP2  JMP  DCCA RA           NF1       JMP IF RESULT POS
3788          7256  35667517777                 SP2  SUB       RA           NCRY
3789          7257  32647767777                 RB   SUB       RB           UNC       NEGATE RESULT
3790          7260  32647377777                 RB   CAD       RB
3791          7261  31766350563                      JMP  DCCA              UNC
3792                                #
3793                                #    SF1, U_ -U, MAY RETURN WITH RANK1 RSB.
3794                                #
3795          7262  31627407477     DM1A        RC   SUB       RC   SF1   ZERO      SF1: LSU_ -LSU, ZERO?
3796          7263  30607307777                 RD   CAD       RD         RSB       NO, MSU_ -MSU-1
3797          7264  30607707777                 RD   SUB       RD         RSB       YES, MSU_ -MSU
3798                                #
3799                                #    F1_ORG USGN=SP2 XOR VSGN, V_ -V, MAY RETURN WITH RANK1 RSB.
3800                                #
3801          7265  35763377312     DM1B RB     SP2  XOR                HBF       F1_VSGN XOR ORG USGN
3802          7266  33667407777                 RA   SUB       RA         ZERO      LSV_ -LSV, ZERO?
```

```
                ADDRESS   CONTENTS      LABEL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1970,  2:08 PM

3803            7267  32647307777             RB   CAD       RB         RSB        NO, MSV_ -MSV-1
3804            7270  32647707777             RB   SUB       RB         RSB        YES, MSV_ -MSV
3805                                *
3806                                *    THIS ROUTINE IS EXECUTED ONLY WHEN BOTH MSU<>0 AND MSV<>0 AND
3807                                *    IS ALWAYS AN OVERFLOW.  IT IS NECESSARY TO MULTIPLY MSU*LSV
3808                                *    TO OBTAIN THE RESULT MODULO 2**32.  IT IS NOT
3809                                *    NECESSARY TO MULTIPLY MSU*MSV BECAUSE THIS CAN HAVE NO
3810                                *    EFFECT IN THE MODULO RESULT.
3811                                *
3812            7271  25772607257   DM2A      SP3  REPN            INCN 20         OVERFLOW CASE; INCN
3813            7272  16774333275       SP0  UBUS MPAD SR1         INCT CTRM       MSU*LSV
3814            7273  37777707617            ADD                   SOV  RSB        SP3=MSW OF PRODUCT; SOV
3815                                *
3816                                *    CHECK FOR -2**31 CASE
3817                                *
3818            7274  32777552777   DM4A      RB   ADD  SL1                 NF1     THROW AWAY SIGN; OVF IF NF1
3819            7275  35773007774       UBUS SP2  IOR                      ZERO
3820            7276  37777777617            ADD                   SOV             OVF IF NOT 2**31
3821            7277  37777707777            ADD                         RSB
3822                                *
3823                                *
3824                                *        DDIV
3825                                *    ENTER FROM DMUL WITH RD,RC=ABS(U), RB,RA=ABS(V),
3826                                *    SP2=ORG MSU, F1=WSGN, SR=4.
3827                                *
3828                                *    IF MSV=0 A SHORTER DIVIDE ALGORITHM CAN BE USED.  IF NOT
3829                                *    THEN V MUST BE NORMALIZED SO THAT BIT 0 OF MSV=1.  U MUST
3830                                *    BE SHIFTED ACCORDINGLY, AND THE REMAINDER MUST ALSO BE
3831                                *    SHIFTED.  A MAXIMUM OF 15 SHIFTS IS REQUIRED WHEN MSV=1.
3832                                *    CTSD CAN BE USED TO NORMALIZE SINCE IT WILL DO A CIRCULAR
3833                                *    SHIFT BASED ON CIR, AND THE BITS SHIFTED OUT WILL ALWAYS
3834                                *    BE ZERO, MAKING IT EQUIVALENT TO A LOGICAL SHIFT, WHICH
3835                                *    IS THE SHIFT NEEDED.  NOTE THAT SP3 IS INVALIDATED BY THE CTSD.
3836                                *
3837            7300  32766007342   DDIV      RB   JMP  DDI3                ZERO    SHORT DIVIDE IF MSV=0
3838            7301  33317777637       RA   ADD       SP1  CLO                    SP1_LSV FOR SHIFT; CLO
3839            7302  32772137777       RB   REPC                      NEG         PERFORM DOUBLE NORMALIZE
3840            7303  16653532277       UBUS CTSD SL1  RB   INCT NEG              SHIFT LEFT (CIRC SHIFT)
3841            7304  01677777777       SP1  ADD       RA
3842            7305  14347377777       CTRL CAD       CTRL                        SET UP FOR SHIFT
3843            7306  16177717637       UBUS ADD       BUS  OPND                   SAVE COUNT IN OPND FOR REM
3844                                *
3845                                *    SHIFT U THEN SAME NUMBER OF PLACES THAT V WAS SHIFTED.
3846                                *
3847            7307  30317777777       RD   ADD       SP1
3848            7310  37532337277            REPC      SP3  INCT CTRM          SP3_0
3849            7311  16760332271       RC   UBUS DASL SL1       INCT CTRM
3850            7312  37337777765       RBUS       ADD  SP0                    U=SP3,SP1,SP0
3851                                *
3852                                *    DIVIDE THE TWO MSW'S OF U BY MSV.  THE QUOTIENT THUS OBTAINED
3853                                *    CAN BE TOO LARGE SINCE LSV WAS NOT CONSIDERED IN THE DIVIDE.
3854                                *    BECAUSE MSV>=100000, THE QUOTIENT CAN BE NO MORE THAN 1 TOO LARGE.
3855                                *
3856            7313  25772577777       SP3  REPN                      21         PERFORM DIVIDE
3857            7314  32764332276       UBUS RB   DVSB SL1       INCT CTRM       QUOTIENT=SP1
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:08 PM

3858            7315  37617573425             RBUS      ADD  SR1  RD   CF2  NF2       REMAINDER=RD,SP0
3859            7316  16611700000                  UBUS ROM       RD   100000         PUT BACK HIGH BIT
3860                                  *
3861                                  *    IT IS NECESSARY TO SUBTRACT LSV*QUOTIENT FROM THE REMAINDER.
3862                                  *    IF IT STILL LEAVES A POSITIVE REMAINDER, THEN WE HAVE THE
3863                                  *    CORRECT QUOTIENT AND REMAINDER.  IF NOT, WE MUST ADD BACK
3864                                  *    ONE COPY OF V TO THE REMAINDER AND DECREMENT THE QUOTIENT
3865                                  *    ACCORDINGLY.  WE ARE GUARANTEED THE QUOTIENT IS NO MORE THAN
3866                                  *    1 TOO LARGE BY THE NORMALIZATION.  IF THE QUOTIENT IS ZERO,
3867                                  *    THE REMAINDER IS THE DIVIDEND AND THIS CHECK MAY BE SKIPPED.
3868                                  *    THE REMAINDER IS A 32 BIT UNSIGNED QUANTITY.  THE SIGN BIT IS
3869                                  *    THE 33RD BIT.  WE CAN TELL IF THIS BIT IS NEGATIVE AFTER THE
3870                                  *    SUBTRACTION BY CHECKING FOR THE ABSENCE OF CARRY OUT FROM THE
3871                                  *    MOST SIGNIFIGANT WORD.
3872                                  *
3873            7317  01526007333           SP1  JMP  DDI4 SP3       ZERO      SP3_QUOT FOR MPAD;SKIP IF 0
3874            7320  37772607777           REPN                     20        MULTIPLY TRIAL QUOTIENT BY
3875            7321  16774333273       RA  UBUS MPAD SR1       INCT CTRM      LOW ORDER BITS AND SUBTRACT
3876            7322  17527377777           SBUS CAD       SP3                 FROM REMAINDER
3877            7323  25327517775           SP0  SP3  SUB       SP0       NCRY      REMAINDER=RD,SP0
3878            7324  25776407777           SP3  INC                      ZERO      SKIP IF CARRY
3879            7325  30617517776           UBUS RD   ADD       RD        NCRY      NEG REMAINDER IF NCRY
3880            7326  37766367333                     JMP  DDI4             UNC       JMP IF CORRECTION NOT REQ'D
3881                                  *
3882                                  *    CORRECTION NECESSARY.  ADD BACK THE DIVISOR AND DECREMENT QUOTIENT.
3883                                  *
3884            7327  33337517775           SP0  RA   ADD       SP0       NCRY      ADD BACK DIVISOR
3885            7330  32616767770           RD   RB   INC       RD        UNC
3886            7331  32617777770           RD   RB   ADD       RD
3887            7332  37307377774           SP1       CAD       SP1                 DECREMENT QUOTIENT
3888                                  *
3889                                  *    NORMALIZE REMAINDER.  IT IS SUFFICIENT TO DO A CTSD (WHICH
3890                                  *    DOES A CIRCULAR SHIFT) BECAUSE THE LOWER BITS OF THE
3891                                  *    REMAINDER WILL ALWAYS BE ZERO SINCE THEY WERE
3892                                  *    ZERO IN THE DIVISOR AND DIVIDEND (BY NORMALIZATION).
3893                                  *    THUS A CIRCULAR SHIFT IS AGAIN EQUIVALENT TO A LOGICAL SHIFT.
3894                                  *    NOTE THAT SP1 IS INVALIDATED BY THE CTSD.
3895                                  *
3896            7333  26357777777  DDI4     OPND ADD       CTRL                SET UP CTR TO SHIFT REM
3897            7334  37537777775       SP0       ADD       SP3
3898            7335  01637777777           SP1  ADD       RC                  RD,RC_QUOTIENT
3899            7336  37617777777                ADD       RD
3900            7337  30652337277           RD   REPC      RB   INCT CTRM      NORMALIZE REMAINDER
3901            7340  16653733277           UBUS CTSD SR1  RB   INCT CTRM      (CIRC SHIFT) LOW SP0=0
3902            7341  25666367355           SP3  JMP  DDI5 RA        UNC       REMAINDER IN RB,RA
3903                                  *
3904                                  *    A SHORT DIVIDE IS POSSIBLE SINCE MSV=0 WHICH GUARANTEES THAT
3905                                  *    THERE ARE NO MORE THAN 16 SIGNIFIGANT BITS IN THE DIVISOR.
3906                                  *    THE DVSB CAN HANDLE UP TO A 16 BIT DIVISOR AND A 32 BIT DIVIDEND.
3907                                  *    IF THE DIVISOR DOES NOT GO THE FIRST TIME, IT IS NOT NECESSARY
3908                                  *    TO DIVIDE 0,MSU BY LSV SINCE THIS WILL BE ZERO.  IF IT DOES
3909                                  *    GO,  JMP OFF TO DIVIDE THE UPPER PORTION AND GET THE MSW
3910                                  *    OF THE QUOTIENT.
3911                                  *
3912            7342  33766003130  DDI5     RA   JMP  TRP4            ZERO      TRAP IF DIV BY ZERO
```

```
                ADDRESS   CONTENTS     LAB  RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:08 PM

3913            7343   37177777637                ADD       RUS  OPND              OPND_MSW QUOTIENT (=0)
3914            7344   33767517770          RD    RA   SUB                    NCRY   NEED TO DIV MSU?
3915            7345   30302367373               RD   JSB  DDI2 SP1          UNC    YES! JMP, SP1_MSU
3916            7346   31317777637               RC   ADD       SP1  CLO            SP1_LSU; CLO
3917            7347   30772577777               RD   REPN                   21
3918            7350   33764332276          UBUS RA   DVSB SL1              INCT CTRM  MSU,LSU/LSV
3919            7351   37677573425          RBUS      ADD  SR1  RA   CF2  NF2       RESTORE HIGH BIT FROM F2
3920            7352   16671700000               UBUS ROM       RA   100000         RB,RA_REMAINDER (RB=0)
3921            7353   01637777777               SP1  ADD       RC                  RD,RC_QUOTIENT
3922            7354   26617777777               OPND ADD       RD
3923                                  *
3924                                  *    COMPLIMENT THE QUOTIENT IF A NEGATIVE RESULT IS EXPECTED.
3925                                  *    OVERFLOW IS POSSIBLE ONLY IN THE CASE OF -2**31/-1.
3926                                  *
3927            7355   37766157362   DDI5                JMP  DDI6                 NF1    JMP IF POS RESULT
3928            7356   31627507777               RC   SUB       RC                 CRRY
3929            7357   30607367777               RD   CAD       RD                 UNC    COMPLIMENT QUOTIENT
3930            7360   30607777777               RD   SUB       RD
3931            7361   37766367364                    JMP  DDI7                 UNC    CAN'T OVF IF NEG
3932            7362   30777527777   DDI6        RD   ADD                       POS    OVF IF -2**31
3933            7363   37777777617                    ADD                  SOV
3934                                  *
3935                                  *    IF THE DIVIDEND WAS NEGATIVE THEN THE REMAINDER MUST ALSO
3936                                  *    BE NEGATIVE.
3937                                  *
3938            7364   35766127370   DDI7        SP2  JMP  DDI8                 POS
3939            7365   33667507777               RA   SUB       RA                 CRRY   COMPLIMENT REMAINDER
3940            7366   32647367777               RB   CAD       RB                 UNC    IF DIVIDEND<0
3941            7367   32647777777               RB   SUB       RB
3942                                  *
3943                                  *    DONE.  SET OCCA ON QUOTIENT IN RD,RC.
3944                                  *
3945            7370   30777407757   DDI8        RD   ADD                  CCA  ZERO   SET OCCA ON QUOTIENT
3946            7371   37777757777                    ADD                       NEXT
3947            7372   31777757657               RC   ADD                  CCZ  NEXT
3948                                  *
3949                                  *    THIS SUBROUTINE IS EXECUTED WHEN THE DIVIDE SUCCEEDS THE FIRST
3950                                  *    TIME.  IT GIVES THE UPPER WORD OF A TWO WORD QUOTIENT.  THE
3951                                  *    REMAINDER IS STORED BACK IN MSU(RD) FOR THE SECOND DIVIDE
3952                                  *    PERFORMED IN LINE.
3953                                  *
3954            7373   37772577777   DDI9                REPN                   21     DIVIDE MS BITS
3955            7374   33764332276          UBUS RA   DVSB SL1              INCT CTRM  0,MSU/LSV
3956            7375   37617573425          RBUS      ADD  SR1  RD   CF2  NF2
3957            7376   16611700000               UBUS ROM       RD   100000         PUT BACK MSB OF REM
3958            7377   01177707637               SP1  ADD       BUS  OPND RSB       OPND_MSW QUOTIENT
```

```
ADDRESS   CONTENTS      LABEL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:08 PM

3959                                &              SECTOR 15
3960                                *
3961                                *          DIO COLD LOAD
3962                                *
3963                                *          DISC DUMP
3964                                *
3965                                *
3965                                *          DIO COLD LOAD USING ATC & 2644
3967                                *     ENTRY FROM DCLD (REG COLD LOAD) WITH SP1=DEV#,
3968                                *     SP2=LABEL, CTR=S-BNK+1, S-BNK=0, F2=0, SR=0.
3969                                *
3970                                *     INITIALIZE ATC (ASSUME I/O RESET)
3971                                *     CONFIGURE FOR SEND @2400 BAUD
3972                                *
3973                                &7473
3974        7473  37431700405      DCL0            ROM         Q     160405          CONFIGURE FOR SEND
3975        7474  37671607476                     ROM         RA    DIO1            RETURN ADR
3976        7475  37506367532                     JMP   TWIO  STA         UNC       STA_01 GO WRITE
3977        7476  37436772057      DIO1           INC   SL1   Q     CLSR            SEND TO UNIT 0
3978        7477  37671607501                     ROM         RA    DIO2            RETURN ADR
3979        7500  37766367545                     JMP   TCIO              UNC       SEND TO CHANNEL 0
3980                                *
3981                                *     CONFIGURE FOR RECEIVE @2400 BAUD
3982                                *
3983        7501  37431740405      DIO2           ROM         Q     120405          CONFIGURE FOR RECIEVE
3984        7502  37671607504                     ROM         RA    DIO3            RETURN ADR
3985        7503  37766367532                     JMP   TWIO              UNC       GO WRITE
3986        7504  37436772777      DIO3           INC   SL1   Q                     SEND TO UNIT 0
3987        7505  37671607507                     ROM         RA    DIO4            RETURN ADR
3988        7506  37766367545                     JMP   TCIO              UNC       SEND TO CHANNEL 0
3989                                *
3990                                *     SEND ESCAPE LOWER CASE e TO READ FROM 2644
3991                                *
3992        7507  37771615545      DIO4           ROM               015545          ESC   LC E
3993        7510  16602367551                UBUS JSB   SEND RD          UNC       SEND ESCAPE
3994                                *
3995                                *     RECEIVE N WORDS AND WRITE TO 0
3996                                *
3997        7511  37417777777                     ADD         P                     WRITE TO 0
3998        7512  37762367573                     JSB   RECV              UNC
3999        7513  32617775777                RB   ADD   RLZ   RD
4000        7514  37762367573                     JSB   RECV              UNC
4001        7515  32777774777                RB   ADD   RRZ
4002        7516  30616777776                UBUS RD   INC         RD              RD_COUNT
4003        7517  37762367573      CLD1           JSB   RECV              UNC       GET A CHARACTER
4004        7520  32637775777                RB   ADD   RLZ   RC                  RC_UPPER BYTE
4005        7521  37762367573                     JSB   RECV              UNC       GET ANOTHER CHARACTER
4006        7522  32657774777                RB   ADD   RRZ   RB                  EXTRACT LOWER BYTE
4007        7523  20177777157                P    ADD         BUS   WRA             WRITE TO MEMORY

4008        7524  31177777432           RB   RC   ADD         BUS   DATA
4009        7525  20416777777           P    INC         P                       INC ADDRESS
4010        7526  37607007450           RD        CAD         RD    CF1   ZERO      DEC PASS COUNT, F1=0
4011        7527  37766367517                     JMP   CLD1              UNC       KEEP READING
4012        7530  35526303033                SP2  JMP   INT2  SP3         UNC       SP3=LABEL, SET UP REGS
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                    FRI, AUG 13, 1976,  2:08 PM

4013                                    *
4014                                    *
4015                                    *    TERMINAL COMMUNICATION ROUTINES
4016                                    *    ASSUME DEVICE NUMBER STORED IN SP1
4017                                    *
4018            7531  01531413400       TRI0      SP1  ROM        SP3  3400 NZRO      ALWAYS NZRO
4019            7532  01531701400       TWI0      SP1  ROM        SP3  101400
4020            7533  01031702400       TRW1      SP1  ROM        IOA  102400        DO TIO
4021            7534  12777532777                 IOD  ADD  SL1             NEG       CHECK BIT 1 (RIO/WIO OK)
4022            7535  37766367533                      JMP  TRW1            UNC       NOT READY
4023            7536  25761412000                 SP3  ROMN               2000 NZRO    SKIP IF TRIO
4024                                    *
4025            7537  21057777777       TIO0      Q    ADD        IOD
4026            7540  25031300000       TIOA      SP3  ROMI       IOA  100000
4027            7541  12537777777                 IOD  ADD        SP3                 UNLOAD DATA TO SP3
4028            7542  04777423777                 CPX1 ADD  SR1             EVEN
4029            7543  37766362762                      JMP  SYSH            UNC       FAIL IF NO RESPONSE
4030            7544  33577777777                 RA   ADD        RAR                 RETURN
4031                                    *
4032            7545  01531700400       TCI0      SP1  ROM        SP3  100400
4033            7546  37766367537                      JMP  TIOD            UNC
4034                                    *
4035            7547  01531702400       TTI0      SP1  ROM        SP3  102400
4036            7550  37766367540                      JMP  TIOA            UNC
4037                                    *
4038                                    *
4039            7551  16437760477       SEN0      UBUS ADD  LRZ  Q    SF1  UNC
4040            7552  30437774457       SEN5      RD   ADD  RRZ  Q    CF1
4041            7553  37671607556       SEN6           ROM        RA   SEN1           RETURN ADR
4042            7554  21431243400                 Q    ROMI       Q    043400         ADD CONTROL BITS
4043            7555  37766367532                      JMP  TWIO            UNC
4044            7556  37436772777       SEN1           INC  SL1  Q
4045            7557  37671607561                      ROM        RA   SEN2           RETURN ADR
4046            7560  37766367545                      JMP  TCIO            UNC
4047            7561  37671607563       SEN2           ROM        RA   SEN3           RETURN ADR
4048            7562  37766367547                      JMP  TTIO            UNC
4049            7563  25761604000       SEN3      SP3  ROMN               004000
4050            7564  16766007561                 UBUS JMP  SEN2            ZERO
4051            7565  37436777777       REC2           INC        Q                   ACK INT
4052            7566  37671607571                      ROM        RA   SEN4           RETURN ADR
4053            7567  25657774777                 SP3  ADD  RRZ  RB                  UNLOAD SP3 IN CASE RECV
4054            7570  37766367545                      JMP  TCIO            UNC
4055            7571  37766147552       SEN4           JMP  SENS            F1
4056            7572  37777707777                      ADD                  RSB
4057                                    *
4058                                    *
4059            7573  37671607575       RECV           ROM        RA   REC1           RETURN ADR
4060            7574  37766367547                      JMP  TTIO            UNC
4061            7575  25761604000       REC1      SP3  ROMN               004000
4062            7576  16766007573                 UBUS JMP  RECV            ZERO
4063            7577  37671607565                      ROM        RA   REC2           RETURN ADR
4064            7600  37766367531                      JMP  TRIO            UNC
4065                                    *
4066                                    *
4067                                    *         7905 DUMP
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:08 PM

4068                                    *    SIO DMP TO 7905 BEG AT CYLINDER %000572, HEAD/SEC %001000;        ###
4069                                    *    SEC AND CYLINDER ARE INCREMENTED SEQUENTIALLY BY DISC CONTROLLER.
4070                                    *    ENTRY FROM DCLD (REG DMP) WITH SP1=DEV#, CTR=S-BNK+1, S-BNK=0, F2=1
4071                                    *
4072            7601  013177?2054       DDM0 SP1  SP1  ADD  SL1  SP1  CLSR          SP1_DEV# * 4, SR_0
4073            7602  37157777017                    ADD       SBR  ABS           ABS-BNK_0
4074                                    *
4075                                    *    LOAD TOS REGS, SP3, SP2, NIR, PCLK AND OPND WITH (S-BNK,SP1)...
4076                                    *    S-BNK=0, SR=0; SP0_SP1+4, SP1_SP1+7, SF1, F3_ICS FLG, SET ICS, SR_4
4077                                    *
4078            7603  01137777777       DPL9      SP1  ADD       BSP0 ROS           READ D0-D4
4079            7604  01316637177                 SP1  INC       SP1  CF3  SRN4     INCR SP1, F3_0
4080            7605  26537767777                 OPND ADD       SP3       UNC      SP3_D4
4081            7606  26206367603                 OPND JMP  DPL9 PUSH      UNC      TOS_D0-D3
4082            7607  04761400020                 CPX1 ROMN              0020 ZERO
4083            7610  37777777077                      ADD       SF3                F3_ICS FLAG
4084            7611  01177777777                 SP1  ADD       BUS  ROS           READ D5
4085            7612  16116777717                 UBUS INC       BSP1 RNS           READ D6 INTO NIR
4086            7613  26737777117                 OPND ADD       SP2  SIFG          SP2_D5, SET ICS FLAG
4087            7614  01116777777                 SP1  INC       BSP1 ROS           READ D7
4088            7615  16176777777                 UBUS INC       BUS  ROS           READ D8
4089            7616  26017777477                 OPND ADD       PCLK SF1           PCLK_D7, F1_1
4090                                    *
4091                                    *    CREATE AND EXECUTE SEEK I/O PROGRAM, BEGINNING AT
4092                                    *      ABS-BNK,SP0-4 THROUGH SP1+1 = DEV#*4 THROUGH DEV#*4+8;
4093                                    *      I/O PTR, CONTROL SEEK, WRITE 2W FROM D7, END, CYLINDER, HEAD/SEC.
4094                                    *    F1=1, F2=1, SR=4; DDBA (DD0C) USED TO WRITE I/O PTR AND SIO;
4095                                    *      RETURN TO DDM1 WITH SP0_SP3, SP1_DEV#*4, SP3 MODIFIED.
4096                                    *
4097            7617  01176777157       DD0A      SP1  INC       BUS  WRA
4098            7620  37171601000                 ROM       BUS  001000        HEAD/SEC                  ###
4099            7621  01177777157                 SP1  ADD       BUS  WRA
4100            7622  37171600572                 ROM       BUS  000572        CYLINDER                  ###
4101            7623  37176777155            SP0       INC       BUS  WRA
4102            7624  37171630000                 ROM       BUS  030000        IOCW (END, INT AFTER SEEK)
4103            7625  37177777155            SP0       ADD       BUS  WRA
4104            7626  01177777437                 SP1  ADD       BUS  DATA          IOAW (D7 ADDR)
4105            7627  37127377155            SP0       CAD       BSP0 WRA
4106            7630  37171667776                 ROM       BUS  067776        IOCW (WRITE 2)
4107            7631  37127377155            SP0       CAD       BSP0 WRA
4108            7632  37171601000                 ROM       BUS  001000        IOAW (SEEK)
4109            7633  37127377155            SP0       CAD       BSP0 WRA           SP0_DEV#*4+1
4110            7634  37171640000                 ROM       BUS  040000        IOCW (CONTROL)
4111            7635  37766367644                 JMP  DD0C            UNC      FINISH SEEK PROGRAM
4112                                    *
4113                                    *    CREATE AND EXECUTE AUTO SEEK I/O PROGRAM, BEGINNING AT
4114                                    *      ABS-BNK,SP1 = DEV#*4; I/O PTR, CONTROL, END WITH INTERRUPT.
4115                                    *    F1=0, F2=1, SR=4; DDBA USED TO WRITE I/O PTR AND SIO;
4116                                    *      RETURN TO DDM1 WITH SP0_SP3, SP1_DEV#*4, SP3 MODIFIED.
4117                                    *
4118            7636  01136777157       DD0B      SP1  INC       BSP0 WRA           SP0_DEV#*4+1
4119            7637  37171640000                 ROM       BUS  040000        IOCW (CONTROL)
4120            7640  37116777155            SP0       INC       BSP1 WRA
4121            7641  37171607405                 ROM       BUS  007405        IOAW (SAME HEAD, AUTO SEEK)
4122            7642  01176777157                 SP1  INC       BUS  WRA
```

```
4123          7643  37171634000                     ROM       BUS  034000          IOCW (END WITH INT)
4124          7644  07301600077  DD0C       SWCH ROMN       SP1  000077          SP1_DEV#, SAVE SP3, FINISH
4125          7645  25326367672                SP3  JMP  DD8A SP0       UNC         SEEK PROGRAMS (1C JMP)
4126                             *
4127                             *    AFTER EXECUTING SEEK I/O PROGRAMS DD0A AND DD0B RETURN TO DDM1 WITH
4128                             *      SP0=ORG (D4), SP1=DEV#*4; SP3_SP0, CF1, EXIT TO DD0B IF F1 WAS ON
4129                             *    CREATE SECOND PART OF WRITE I/O PROGRAM, BEGINNING AT
4130                             *      ABS=BNK,SP1+3 = DEV#*4+3; CTRL WRT, WRT 4K FROM D6, END W/ INT.
4131                             *    SR=4; ORG (D8) RESTORED FROM OPND, RH SP0_0.
4132                             *
4133          7646  37537557455  DDM1 SP0        ADD       SP3  CF1  NF1
4134          7647  37766367636                JMP  DD0B                UNC
4135          7650  01137777141       SR   SP1  ADD       BSP0 WRA
4136          7651  37171604000                 ROM       BUS  004000          IOAW (WRITE)
4137          7652  37167377155       SP0       CAD       BUS  WRA
4138          7653  37171640000                 ROM       BUS  040000          IOCW (CONTROL)
4139          7654  37136777155       SP0       INC       BSP0 WRA
4140          7655  37171600000                 ROM       BUS  060000          IOCW (WRITE 4K)
4141          7656  37136777155       SP0       INC       BSP0 WRA
4142          7657  16177777437            UBUS ADD       BUS  DATA            IOAW (D6 ADDR)
4143          7660  37136777155       SP0       INC       BSP0 WRA
4144          7661  37131634000                 ROM       BSP0 034000          IOCW (END WITH INT)
4145          7662  37176777155       SP0       INC       BUS  WRA                       \RH SP0_0
4146          7663  26177777437            OPND ADD       BUS  DATA            RESTORE ORG (D8)
4147                             *
4148                             *    CREATE FIRST PART OF WRITE I/O PROG, BEGINNING AT ABS=BNK,DEV#*4;
4149                             *      I/O PTR, SET BNK=RH SP0; SIO; SEND RIL TO INTERRUPTING DEVICES,
4150                             *      WAITING FOR CORRECT DEVICE; CHECK I/O PTR IF DMP REC 1-N (SRN4),
4151                             *    SP0_SP3, SP1_DEV#*4+7, SP3_SM, F2_0; EXIT TO DDM4 IF DMP REC 1-N.
4152                             *    DD0A AND DD0B USE DD8A TO WRITE I/O PTR AND EXECUTE I/O PROGRAMS;
4153                             *      SP0=DEV#*4+1 ON ENTRY (UNCH BY DD8A), SP1=DEV#, F2=1, SR=4;
4154                             *      EXIT TO DDM1 WITH SP1=DEV#*4, SP3 MODIFIED.
4155                             *
4156          7664  07301600077  DDM8       SWCH ROMN       SP1  000077          SP1_DEV#
4157          7665  16136772156       UBUS UBUS INC  SL1  BSP0 WRA
4158          7666  37177774435       SP0       ADD  RRZ  BUS  DATA            IOAW (BANK=SP0(14:15))
4159          7667  37127377155       SP0       CAD       BSP0 WRA
4160          7670  37171614000                 ROM       BUS  014000          IOCW (SET BANK)
4161          7671  25337777437            SP3  ADD       SP0  CF2             SAVE SP3, F2_0
4162          7672  37107377155  DD8A SP0        CAD       BSP1 WRA             AT DEV# * 4,
4163          7673  01531701000            SP1  ROM       SP3  101000            (FORM SIO CMD)
4164          7674  01176777437            SP1  INC       BUS  DATA              PUT DEV# * 4 + 1
4165          7675  37762351724  DDM9           JSR  IOPA            UNC         SEND CMD TO DEVICE
4166          7676  37511644000                 ROM       STA  044000          SET I BIT, AND K
4167                             *                                                 FOR CLEARING EXT INT
4168          7677  24537777777            STA  ADD       SP3                  SAVE STA
4169          7700  04777467777            CPX1 ADD                 BITR
4170          7701  37766367700                 JMP  *-1                UNC         WAIT FOR INT
4171          7702  25517777777            SP3  ADD       STA                  REPLACE STA
4172          7703  24777627037            STA  ADD            CCPX SR4        CLEAR EXT INTERRUPT
4173          7704  01177777177            SP1  ADD       BUS  ROA             READ I/O PTR IF DMP 1-N
4174          7705  11531302000            IOA  ROMI      SP3  102000          FORM RIL CMD
4175          7706  07763377776       UBUS SWCH XOR
4176          7707  16761400077            UBUS ROMN            0077 ZERO
4177          7710  37766367675                 JMP  DDM9               UNC         JMP IF NOT CORRECT DEVICE
```

```
4178          7711  37762301724                 JSB  IOPA            UNC      SEND RIL TO (CORR) DEVICE
4179          7712  23526167646           SM    JMP  DDM1 SP3       F2       SP3_SM: JMP IF SEEK PROGS
4180          7713  01311600007           SP1   ROM       SP1  000007        SP1_DEV##4+7
4181          7714  26767537776     UBUS  OPND  SUB                 NEG      JMP IF ABN END AND DMP 1-N
4182          7715  37766237751                 JMP  DDM5            SRN4           \(FOLLOWING SBUS=0)
4183          7716  37766237745                 JMP  DDM4            SRN4     JMP IF DMP REC 1-N
4184                                *
4185                                *    PUSH REGS,ETC, INCL D0-5,7 IN TOS,SP0,SP2,PCLK, ORG S-BNK(+1)
4186                                *      IN CTR AND ORG SM IN SP3 EXCEPT CPX1,CPX2,D6 AND SIZE INTO
4187                                *      MEM BEGINNING AT S-BNK,SP1+1=DEV##4+8.
4188                                *    SR=4; SP2_4K, RD_0, RB_CPX1 INCL F3, SR_2, SM_DEV##4+28.
4189                                *
4190          7717  01462211744           SP1  JSB  PSHA SM        SRNZ     DUMP D0-D3 (2C JMP)
4191          7720  37217777775     SP0         ADD       PUSH
4192          7721  35217777777           SP2  ADD       PUSH
4193          7722  13202301744           PCLK JSB  PSHA PUSH      UNC      DUMP D4,D5,D7
4194          7723  37217777766     X           ADD       PUSH
4195          7724  34217777777           DL   ADD       PUSH
4196          7725  03217777417           RBR  ADD       PUSH DB
4197          7726  22202301744           DB   JSB  PSHA PUSH      UNC      DUMP X,DL,DB-BANK,DB
4198          7727  21217777777           Q    ADD       PUSH
4199          7730  25202301744           SP3  JSB  PSHA PUSH      UNC      DUMP Q,SM
4200          7731  14211777777           CTRL ROM       PUSH 177777        (CTR=S-BANK+1)
4201          7732  37217777762     Z           ADD       PUSH
4202          7733  24202301744           STA  JSB  PSHA PUSH      UNC      DUMP S-BANK,Z,STA
4203          7734  04217777217           RBR  ADD       PUSH PB
4204          7735  36202301744           PB   JSB  PSHA PUSH      UNC      DUMP PB-BANK,PB
4205          7736  20217777777           P    ADD       PUSH
4206          7737  37217777760     PL          ADD       PUSH
4207          7740  00202301744           CIR  JSB  PSHA PUSH      UNC      DUMP P,PL,CIR
4208          7741  37731610000                ROM       SP2  010000        SP2_4K
4209          7742  37617777217                ADD       RD   INSR         RD_0, SR=1
4210          7743  04657747217           CPX1 ADD       RB   INSP F3      RB_CPX1, SR=2,
4211          7744  16641777757           UBUS ROMN      RB   177757          ICS FLAG WAS OFF
4212                                *
4213                                *    FIND MEM SIZE; PUSH CPX1,CPX2,D6 AND SIZE INTO MEM BEGINNING AT
4214                                *      S-BNK,SM+1=DEV##4+29; DUMP REC 1-N (4K EACH, BEG WITH 0-7777).
4215                                *    ENTER WITH RD=0, RB=CPX1, SR=2, SM=DEV##4+28, ABS-BNK=0, S-BNK=0,
4216                                *      SP1=DEV##4+7, SP2=4K, SP3=ORG SM, NIR=D6.
4217                                *    DDM8 USED TO WRITE REC 1-N; WRITE 4K AND WRITE ADDR SET AT
4218                                *      DEV##4+5,6, RH SP0_BNK AND SM_SP3 BEFORE EACH TRANSFER TO DDM8;
4219                                *      IF ABN I/O PROGRAM END DDM8 ENTERS AT DDM5 WITH SBUS=0 (SR=0)
4220                                *      TO TERMINATE DMP WITH ADDR+BNK OF REC BEING WRITTEN IN CIR.
4221                                *    EXIT TO WAIT AFTER LAST RECORD, WITH ENVIRONMENT UNCHANGED EXCEPT
4222                                *      (DEV##4) THROUGH (DEV##4+32), CIR=LAST ADDR+1 + S-BNK=BNK.
4223                                *
4224          7745  35617417770     DDM4  RD    SP2  ADD       RD       NZRO     INC MEM ADDR
4225          7746  03156777617                RBR  INC       SBR  S           INC S-BANK
4226          7747  16137617770           RD   UBUS ADD       BSP0 ROS SRNZ    READ (ADDR+S-BNK)
4227          7750  16177777237                UBUS ADD       BUS  NIR         NIR_ADDR+BNK IF DMP 1-N
4228          7751  17631601200     DDM5       SBUS ROM       RC   001200      RC_ADDR+BNK+1200
4229          7752  37777777025           RBUS      ADD            CCPX        CLR ILLEGAL ADDR, CIR_NIR
4230          7753  04621620000                CPX1 ROMN      RC   020000      TEST FOR ILLEGAL ADDR INT
4231          7754  31761404003                RC   ROMN           4003 ZERO   256K?
4232          7755  31766007764                RC   JMP  DDM6           ZERO     JMP IF NOT ILLEGAL ADDR
```

```
          ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, AUG 13, 1976,  2:08 PM

4233        7756  06677777777                CPX2 ADD       RA                RA_CPX2
4234        7757  37626202761                     JMP  WAIT RC        SRZ      RC_0; JMP IF ALL DUMPED
4235        7760  37157777617                     ADD       SBR  S            RESET S-BANK
4236        7761  00202361744                CIR  JSB  PSHA PUSH      UNC      RD_RC; DUMP CPX1,CPX2,D6
4237        7762  37677777215           SP0       ADD       RA   INSR         SIZE=ADDR+BANK
4238        7763  37322361744                     JSB  PSHA SP0       UNC      CLR SP0; DUMP SIZE
4239        7764  37766217745     DDM4            JMP  DDM4                SRN7 JMP IF FINDING SIZE
4240        7765  37177777235           SP0       ADD       BUS  NIR          NIR_ADDR+BNK
4241        7766  37107377154           SP1       CAD       BSP1 WRA
4242        7767  30177777437                RD   ADD       BUS  DATA         SET WRITE ADDR
4243        7770  37167377154           SP1       CAD       BUS  WRA
4244        7771  37171660000                     ROM       BUS  060000       WRITE 4K
4245        7772  25466367664                SP3  JMP  DDM8 SM        UNC      SET I/O PTR,BANK; SIO
4246                            *
4247                            *
4248        7773  37766177473     DCLD            JMP  DCL0                NF2  DIRECT CLD LD/DMP ENTRY
4249        7774  37766367601                     JMP  DDMP                UNC
4250        7775  37766367123     NQT1            JMP  NQT2                UNC  FP N**2 ENTRY
4251        7776  37777777777                     ADD

4252        7777  37777777777     UNIV            ADD
4253                            *72,2 1K PARITY
4254                            *7472 1K PARITY
4255                            *7472 2K PARITY
4256                            $STAT
                                *

                ROM COUNT=2412    ERRORS=0        WARNINGS=40
```

```
AC12  0044  <=  0074
AC13  0050  <=  0057
AC14  0055  <=  0047  0141
AC1D  0043
AC1P  0060
AC1S  0042
AC2D  0104
AC2P  0114
AC2S  0103
AC3D  0126
AC3S  0125
AC4D  0154
AC4S  0153
AC5D  0166
AC5S  0165
ADAX  0605
ADBX  0611
ADD   0612
ADDI  0760
ADDM  0075
ADDS  1553
ADXA  0603
ADXB  0607
ADXI  0754
AINC  0070

ALS1  0226  <=  0255
ALS2  0253  <=  0233
ALS3  0256  <=  0230
```

```
ALSB  0222

AND   0033

ANDI  0757

ASWK  1716  <=  1617  1630  1641  1653  1660  1664  1670

AT10  3712  <=  3731  3751

ATS1  3716  <=  3777

ATS2  3727  <=  3742

ATS3  3742  <=  3775

ATS4  3747  <=  3776

ATS5  3757  <=  7140

ATS6  3762  <=  3710

ATS7  3766  <=  3756

ATS8  3771  <=  3735

ATST  3777  <=  3525

BCC1  0422  <=  0402

BCC2  0424  <=  0436  0441  0444  0446  0451  0454  0457  0462  0465  0470  0500  0517

BCC3  0427  <=  0433

BCC4  0432  <=  0411

BCY   0445

BNCY  0443

BND2  1752  <=  1451

BNDC  1751  <=  1533  1541  1557  2452

BNDV  3013  <=  2131  2220  2446

BNOV  0440

BOV   0434

BRD   0401

BRE   0464

BRO   0467
```

```
BRP   0412
BRP2  0414  <=  0421
BRS   0400
BTST  1262
CAB   0577
CIO   1660
CLA2  2711  <=  2407  2441  3112
CLAB  2705  <=  2404  3110
CLD1  7517  <=  7527
CMD   1670
CMP   0615
CMPO  2132  <=  2065
CMPB  2135  <=  2155
CMPI  0537
CMPM  0004
COLD  3206  <=  3154
CPRB  0472
CPRS  2765  <=  0002  2751
CSTV  3122  <=  2524  2717
D031  2202  <=  2001  2024  2071  2135  2160  2207  2226
D03S  2201  <=  2125  2215
D05S  2206  <=  2244  2264  2276
DABZ  0461
DADD  0620
DBBC  2321  <=  2030  2033  2052  2064  2167
DBWC  2313  <=  2016  2020
DCCA  0563  <=  0576  0623  0627  0633  1174  1272  1275  1300  1303  7255  7261
DCLO  7473  <=  7773
```

```
DCLD  7773  <=  3230
DCM2  0636  <=  1203  1206
DCMP  0634
DD0A  7617
DD0B  7636  <=  7647
DD0C  7644  <=  7635
DD8A  7672  <=  7645
DDEL  0644  <=  2204
DDI2  7373  <=  7345
DDI3  7342  <=  7300
DDI4  7333  <=  7317  7326
DDI5  7355  <=  7341
DDI6  7362  <=  7356
DDI7  7364  <=  7361
DDI8  7370  <=  7364
DDIV  7300  <=  7235
DDM1  7646  <=  7712
DDM4  7745  <=  7716  7764
DDM5  7751  <=  7715
DDM6  7764  <=  7755
DDM8  7664  <=  7772
DDM9  7675  <=  7710
DDMP  7601  <=  7774
DDUP  0763
DECA  0555
DECB  0557
DECX  0553
DEL   0645  <=  0220
```

```
DELB  0646
DEXF  1400
DFL2  1221  <=  1215
DFLT  1212
DIO1  7476
DIO2  7501
DIO3  7504
DIO4  7507
DISP  2643  <=  1545
DIV   0721
DIVI  0542
DIVL  0724
DM1A  7262  <=  7233
DM1B  7265  <=  7234
DM2A  7271  <=  7246
DM4A  7274  <=  7254
DMEM  3172  <=  3162
DMP1  3244  <=  3234
DMP3  3336  <=  3521
DMP4  3341  <=  3313  3360
DMP5  3345  <=  3311
DMP6  3360  <=  3351
DMP8  3261  <=  3366
DMP9  3272  <=  3305
DMU2  7243
DMU3  7247  <=  7243
DMU4  7254  <=  7247
DMUL  7233  <=  3470  3474
```

```
DNEG  0630
DPF   1412  <=  1406
DPL9  7603  <=  7606
DSEG  2355  <=  2247  2305  2311
DSQ2  2372  <=  2367
DSP2  2653  <=  2670
DSTV  3121  <=  2361  2362
DSUB  0624
DTST  0560  <=  0714  1242  1246
DUMP  3221  <=  3152  3206
DUP   0761
DVL2  0732  <=  0723  0726
DVNR  1651  <=  1622  1635  1645
DXBZ  0453
DXCH  0574
DZR2  0772  <=  0767
DZRO  0767
EX10  2522  <=  2723
EX11  2527  <=  1743  1752  2416  2440  2470
EXF   1407
EXI0  2455  <=  2460
EXI1  2463  <=  2600
EXI2  2471
EXI3  2476  <=  2472
EXI8  2520  <=  3107
EXI9  2521  <=  2711
EXIT  2456
EXSW  3203  <=  3164
```

```
FAD1  1011  <=  1006
FAD4  1031  <=  1025
FADD  1001
FCMP  1201
FDIV  1110
FDV2  1143  <=  1136
FDV3  1163  <=  1151
FDZR  1166  <=  1112
FIX2  1242  <=  1236  1237  1254
FIX4  1247  <=  1231
FIXR  1223
FIXT  1222
FLT   1207
FMPY  1060
FNEG  1175
FNG2  1200  <=  1040  1214  1233
FOV   1056  <=  1053
FSUB  1000
GSCB  2217  <=  2173  2212
HALT  2757
HMOD  3143  <=  3001
IABZ  0456
IDM2  0015  <=  0011
IDMY  0011
INCA  0554
INCB  0556
INCX  0552
INTO  3020  <=  2534
```

```
INT1  3024  <=  2653  3005  3014
INT2  3033  <=  2754  3312  7530
INT3  3045  <=  3036
INT4  3055  <=  2551  3040  3044
INT5  3067  <=  2526  2532  3056  3114  3142
INT6  3071  <=  3066
INT7  3140  <=  2373  3023  3126  2373  2376
IOPA  1724  <=  1621  1626  1633  1637  1643  1656  1666  2550  3272  3306  1765  7675  7711
IOPD  1723  <=  1647  1662  1712
IR    3001  <=  0003  2771
IRD   3000  <=  1603  2023  6073  2126  2136  2225  2265
IX2A  2775  <=  2557
IXBZ  0450
IXI1  2552  <=  2546
IXI2  2554  <=  2542  2611
IXI3  2572  <=  2553  2606  2612
IXI4  2575  <=  2615
IXI6  2601  <=  2552  3070
IXIT  2535
L200  6757  <=  3420
L202  6761  <=  3421
L204  6763  <=  3422
L206  6765  <=  3423
L210  6767  <=  3424
L212  6771  <=  3425
L214  6773  <=  3426
L216  6775  <=  3427
LADD  0647
```

```
LADR  3167  <=  3160
LCK1  2623  <=  2626
LCK2  2641  <=  2630
LCMP  0642
LDB   0236
LDD   0142
LDD2  0150  <=  0306  0334  0766
LDI   0751
LDIV  0663
LDPB  0277
LDV2  0674  <=  0666
LDX   0021
LDXA  0601
LDXB  0606
LDXI  0753
LLBL  2400
LLS1  1574  <=  1605
LLSH  1570
LMEM  3171  <=  3161
LMPY  0655
LOAD  0101
LRA   0123
LREG  3166  <=  3156
LSA5  0335  <=  0325
LSA6  0344  <=  0337
LSAB  0323
LST   0347
LSUB  0650
```

```
MAB1  2301
MABS  2266
MB10  2066  <=  2045
MB20  2070  <=  2123
MB21  2110  <=  2127
MB22  2111  <=  2100
MB24  2124  <=  2106  2110
MB26  2127  <=  2107
MDS   2267
MDS1  2304  <=  2300
MFD2  2253  <=  2312
MFD3  2254  <=  2303
MFDS  2252
MFTD  2240  <=  2163
MFY   0704
MFYI  0701
MFYL  0705
MFYM  0702
MTB2  0512  <=  0532  0536
MTB4  0520  <=  0511  0535
MTB6  0523  <=  0505
MTBI  0502
MTD2  2263  <=  2257
MTDS  2260  <=  2251
MVB3  2063  <=  2053
MVB5  2065  <=  2050
MVBD  2046
MVBL  2226  <=  2275
```

```
MVBP  2047
MVBW  2025
MVW1  2003  <=  2051
MVW2  2016  <=  2007
MVW3  2020  <=  2015
MVW4  2021  <=  2231
MVW5  2023  <=  2237
MVWD  2000
MVWP  2001
MVWS  2353  <=  2022  2235  2255  2262
MW11  2345  <=  2354
MZR0  3207
MZR1  3210
MZR2  3213  <=  3215  3217
NEG   0614
NOP   0564  <=  2636
NORM  1041  <=  1107  1165  1221
NOT   0654
NQ01  7000  <=  7125
NQ10  7016  <=  7117
NQ11  7020  <=  7025
NQ12  7021  <=  7030  7103
NQ13  7026  <=  7107
NQ14  7043  <=  7058
NQ15  7054  <=  7127
NQ23  7062  <=  7041
NQ24  7063  <=  7076
NQ25  7075  <=  7130
```

```
NQ30  7102  <=  7061
NQ40  7110  <=  7158
NQ41  7125  <=  7156
NQ80  7134  <=  7045  7060  7066  7101
NQ81  7133  <=  7056  7075
NQ82  7132  <=  7053  7074
NQ83  7126  <=  7050  7072
NQ85  7145  <=  7124
NQ86  7157  <=  7142  7143  7144  7145  7150
NQ87  7167  <=  7141  7162
NQ88  7170  <=  7152  7154  7201
NQ89  7163  <=  7175
NQ90  7171  <=  7173
NQT1  7775  <=  3526
NQT2  7123  <=  7775
NQTS  7121
OPTX  1613
OR    0023
ORI   0755
PAJT  3520  <=  3225
PAUS  2764  <=  2640
PCAL  2412
PCL0  2433  <=  2412
PCL1  2417  <=  2437
PCL2  2441  <=  2417
PCL3  3077  <=  2421
PCL5  2422  <=  2443  2516  3115  3116
PCL6  2423  <=  2511
```

```
PCN   2616 <= 2535

PIOT  3702 <= 3527  3711

PLSA  0307

PRG1  3530 <= 3671

PRGS  3672 <= 3576  3600  3602  3604  3606  3610  3612  3614  3616  3621  3622  3623  3624  3625  3626  3627  3630  3631  3632
              3633  3634  3646  3647  3650  3651  3654  3657  3662  3665  3670

PRGT  3527 <= 3520

PSD2  2670 <= 2665

PSDE  2655 <= 2647

PSH2  1470 <= 1474

PSHA  1744 <= 1446  1461  1464  1542  1554  1555  1747  2414  2435  2450  2455  2766  3024  3140  3314  3317  3323  3325  3330
              3332  3335  3355  3357  7717  7722  7726  7730  7733  7735  7740  7761  7763

PSHM  1737 <= 0101  0123  0144  0146  0236  0302  0304  0315  0326  0330  0344  0356  0524  0601  0702  0751  0761  0763  0765
              0770  0771  0773  1210  1411  1631  1651  1654  1701  2002  2025  2027  2047  2164  2166  2227  2403  2617

PSHR  1446

PSTA  0317 <= 0314

PUL1  1732 <= 0005  0075  0200  0211  0244  0320  0335  0340  0502  0503  0523  0703  1517  1522  1525  1531  1537  2201  2240
              2241  2433

PUL8  3232 <= 3237

PW2   2755 <= 2427

PWR   2742 <= 0001  3010  3143

QALR  1336

QLR2  1345 <= 1355

QLR3  1346 <= 1340

QLR5  1353 <= 1341

REC1  7575

REC2  7565

RECV  7573 <= 7512  7514  7517  7521  7576

RIO   1630

RMSK  1701
```

```
RSW   1611  <=  1570
SCAL  2411
SCAN  1422
SCN2  1430  <=  1422
SCU   2161
SCU1  2212  <=  2172  2216
SCW   2162
SCW1  2173  <=  2175
SED   1674
SEN1  7556
SEN2  7561  <=  7564
SEN3  7563
SEN4  7571
SEN5  7552  <=  7571
SEN6  7553
SEND  7551  <=  7510
SET1  1507  <=  1500
SET2  1515  <=  1512
SET3  1526  <=  1506
SET4  1535  <=  1530
SET5  1536  <=  1543
SETR  1476
SHDL  1270
SHDR  1276
SHFL  1263
SHFR  1255
SIN   1664
SING  3202  <=  3151  3163
```

```
SIO   1617
SIO2  1764  <=  1626
SMSK  1706
SRP1  0034
SRP2  0030
SRP3  0024
SRP4  0020
SSEG  2712  <=  2512  3103
SST   0360
STAX  0604
STB   0242  <=  0235
STBX  0610
STD   0200
STMK  2672  <=  2420  2520  3025  3141
STOP  2760  <=  3002  3006  3165
STOR  0211  <=  0205
STR2  0221  <=  0247  0317
STTV  3123  <=  2525
STUN  3120  <=  0037  1734  1754  2475
SUB   0613
SUBS  1552
SXIT  2444
SYSH  2762  <=  1730  2523  2531  2635  2663  7543
TAS2  1313  <=  1306  1322
TASL  1304  <=  1317
TASR  1317
TBC   1440  <=  1432  1434  1436
TCBC  1436
```

```
TCIO  7545 <= 7500 7506 7560 7570
TEST  1261
TIO   1653
TIOA  7540 <= 7550
TIOD  7537 <= 7546
TNS2  1330 <= 1324
TNSL  1323
TRIF  1251 <= 3007
TRSC  1432
TRIO  7531 <= 7600
TRP0  3134
TRP1  3133 <= 1251
TRP2  3132 <= 1056
TRP3  3131 <= 1057
TRP4  3130 <= 0542 0664 0733 7342
TRP5  3127 <= 1170
TRP6  3117 <= 0307 0323 0354 0365 1471 1501 1544 1571 1674 1706 1716 2243 2267 2502 2514 2545 2620 2757 2764
TRP7  3124 <= 0000
TRW1  7533 <= 7535
TSSC  1434
TSC1  0274 <= 0313
TSCK  0272 <= 0055 0111 0132 0147 0161 0174 0206 0217 0257 0407
TTIO  7547 <= 7562 7574
TWIO  7532 <= 7475 7503 7555
UANS  1171 <= 1011 1023 1062 1113
UNIM  7777 <= 1613 3400 3401 3402 3403 3500
UNPK  1012 <= 1063 1115
WAIT  2761 <= 3170 3201 3353 3550 3701 3705 3716 7757
```

WIO 1641

XAX 0566

XBX 0570

XCH 0572

XCHD 1544

XEQ 1561

XOR 0027

XORI 0756

ZER2 0774 <= 0772

ZERO 0773

ZROB 0640

ZROX 0775

<<<<<<<<<RBUS>>>>>>>>>

	0017	1754	MREG	0003	3	PADD	0004	17	PL	0000	18	RA	0013	56	RB	0012	43
RBUS	0005	23	RC	0011	27	RD	0010	41	SP0	0015	145	SP1	0014	71	SR	0001	27
UBUS	0016	110	X	0006	35	XC	0007	20	Z	0002	16						

<<<<<<<<<SBUS>>>>>>>>>

	0037	747	CC	0027	2	CIR	0000	30	CPX1	0004	26	CPX2	0006	8	CTRH	0015	11
CTRL	0014	35	DB	0022	42	DL	0034	34	IOA	0011	3	IOD	0012	5	MOD	0005	8
OPND	0026	133	P	0020	19	PADD	0002	88	PB	0036	27	PCLK	0013	3	Q	0021	18
QDWN	0010	2	RA	0033	150	RB	0032	107	RBR	0003	45	RC	0031	53	RD	0030	77
SBUS	0017	19	SM	0023	53	SP1	0001	145	SP2	0035	87	SP3	0025	119	STA	0024	30
SWCH	0007	14	UBUS	0016	266												

<<<<<<<FUNCTION>>>>>>>

ADD	0037	801	AUDO	0033	9	AND	0007	17	BNDT	0015	43	CAD	0016	123	CADO	0012	9
CAND	0005	7	CRS	0032	21	CTSD	0027	4	CTSS	0034	2	DCAD	0036	0	DVSB	0010	10
INC	0035	163	INCO	0031	8	IOR	0026	34	JMP	0014	398	JSB	0004	219	MPAD	0030	12
PNLR	0020	4	PNLS	0021	1	QASL	0000	6	QASR	0001	6	REPC	0024	28	REPN	0025	23
ROM	0023	188	ROMI	0022	27	ROMN	0003	70	ROMX	0002	10	SUB	0017	111	SUBO	0013	5
UBNT	0011	10	XOR	0006	37												

<<<<<<<<<STORE>>>>>>>>>

	0037	962	BSP0	0005	57	BSP1	0004	28	BUS	0007	217	CTRH	0017	16	CTRL	0016	47
DB	0022	7	DL	0034	11	IOA	0001	6	IOD	0002	6	MREG	0003	4	P	0020	20
PB	0036	4	PCLK	0000	3	PL	0011	4	PUSH	0010	85	Q	0021	19	QUP	0013	5
RA	0033	103	RAR	0027	47	RB	0032	106	RC	0031	41	RD	0030	54	SBR	0006	54
SM	0023	22	SP0	0015	65	SP1	0014	154	SP2	0035	93	SP3	0025	111	STA	0024	22
X	0026	27	Z	0012	6												

<<<<<<<<<SKIP>>>>>>>>>

	0037	781	BIT6	0005	12	BIT8	0006	15	CRRY	0010	19	CTRM	0033	55	EVEN	0002	21
F1	0014	34	F2	0016	25	F3	0034	14	INDR	0024	6	JLUI	0031	17	NCRY	0011	43
NEG	0013	50	NEXT	0035	159	NF1	0015	33	NF2	0017	17	NOFL	0007	8	NPRV	0026	28
NSME	0004	8	NZRO	0001	81	ODD	0003	12	POS	0012	47	RSB	0030	46	SR4	0022	29
SRL2	0025	4	SRL3	0027	10	SRN4	0023	11	SRNZ	0021	22	SRZ	0020	16	TEST	0032	12
UNC	0036	407	ZERO	0000	78												

<<<<<<<<<SPECIAL>>>>>>>>>

	0037	1267	CCA	0036	42	CCB	0000	5	CCE	0035	12	CCG	0034	8	CCL	0033	6
CCPX	0001	28	CCRY	0025	7	CCZ	0032	9	CF1	0022	26	CF2	0021	14	CF3	0007	2
CLIB	0016	11	CLO	0031	12	CLSR	0002	16	CTF	0006	23	DCSR	0011	15	FHB	0015	5
HBF	0014	20	INCN	0012	9	INCT	0013	63	INSR	0010	16	LBF	0017	9	POP	0027	36
POPA	0026	14	SCRY	0024	5	SDFG	0005	1	SF1	0023	21	SF2	0020	15	SF3	0003	9
SIFG	0004	3	SOV	0030	8												

<<<<<<<<<MCU>>>>>>>>>

ABS	0000	34	CMD	0002	3	CRL	0001	3	DATA	0021	45	DB	0020	30	DPOP	0022	8
NIR	0011	15	OPND	0031	11	PB	0010	8	RND	0024	0	RNP	0014	5	RNS	0034	3
ROA	0007	53	ROD	0027	18	ROND	0023	0	RONP	0013	2	RONS	0033	0	ROP	0017	9
ROS	0037	31	ROSA	0005	1	ROSD	0025	0	S	0030	23	WRA	0006	54	WRD	0026	7
WRS	0036	11															

<<<<<<<<<SHIFT>>>>>>>>>

	0007	1544	LLZ	0001	13	LRZ	0000	16	RLZ	0005	16	RRZ	0004	47	SL1	0002	81
SR1	0003	64	SWAB	0006	8												

HP 3000 SERIES II COMPUTER SYSTEM

EXTENDED INSTRUCTION SET (EIS)

6-15-76

```
1                                *    MC3000/II FIS REV. A
2                                *
3                                *
4                                *         FADD/FSUB, FMPY/FDIV, FNEG/FCMP
5                                *
6                                *
7                                *         FADD/FSUB, FMPY/FDIV
8                                *    ENTER VIA JMP TABLE IN SEC 7 WITH SR=4,
9                                *    RC,RB,RA=W,U,V OR REL ADDRS; W_U OR V, S_S-3.
10                               *
11                               *         FADD/FSUB
12                               *    CK U,V,W BOUNDS; FETCH U,V, EXCH IF NEC SO THAT ABS(U)>=ABS(V);
13                               *    EMPTY RD AND SAVE W ABS ADDR AT SM+1; SF2 IF ABS(U)=ABS(V);
14                               *    RB,SP2,SP0,RA_U, RD,SP3,SP1,RC_V, CLO, F1,F3 MODIFIED.
15                               *    FAS1,2 USED BY FMPY/FDIV TO CK U,V,W BOUNDS AND CALC W ABS ADDR.
16                               *
17                               &0023 (10023)
18        0023  22117777573      FASB RA   DB   ADD        RSP1 ROD              READ V1, SP1_V ABS ADDR
19        0024  23662360770            SM   JSB  PSPM RA         UNC            RA_SM; EMPTY RD
20        0025  01766777773            RA   SP1  RNDT                           SM>=V?
21        0026  01136777577            SP1  INC        RSP0 ROD                 READ V2, SP0_V ABS ADDR+1
22        0027  26617777777            OPND ADD        RD                       SAVE V1
23        0030  34766777774      FAS1  SP1  DL   RNDT                           V>=DL?
24        0031  22637777771            RC   DB   ADD        RC                  RC_W ABS ADDR
25        0032  16766707773            RA   UBUS RNDT                RSB       SM>=W? RET IF SUBR
26        0033  22117777572            RB   DB   ADD        RSP1 ROD            READ U1, SP1_V ABS ADDR
27        0034  26537777777            OPND ADD        SP3                      SAVE V2
28        0035  34766777771      FAS2  RC   DL   RNDT                           W>=DL?
29        0036  34766777774            SP1  DL   RNDT                           U>=DL?
30        0037  01766707773            RA   SP1  RNDT                RSB       SM>=U? RET IF SUBR
31        0040  01176777577            SP1  INC        BUS  ROD                 READ U2
32        0041  26657777317            OPND ADD        RB   HFF                SAVE U1, F1_USGN
33        0042  02777427777            RADD ADD                  EVEN
34        0043  30611700000            RD   POM        RD   100000             COMPL VSGN IF ESUB
35        0044  30737777337            RD   ADD        SP2  FHB                 SP2_V1 WITH USGN
36        0045  01117777561            SR   SP1  ADD        RSP1 ROD            READ U4
37        0046  26677777777            OPND ADD        RA                       SAVE U2 (TEMP IN RA)
38        0047  35767417152            RB   SP2  SUB             CTE  NZRO       CMP U1,2 WITH V1,2
39        0050  25767407153            RA   SP3  SUB             CTE  ZERO         SF1 IF U1,2>=V1,2
40        0051  37777777077                 ADD                 SF3                SF3 IF U1,2<>V1,2
41        0052  37167377574            SP1       CAD        BUS  ROD              READ U3
42        0053  26677557777            OPND ADD        RA             NF1         SAVE U4
43        0054  33726360057            RA   JMP  FAS4 SP2            UNC         SAVE U2
44        0055  16537777777            UBUS ADD        SP3                      EXCH U2 AND V2
45        0056  25737777777            SP3  ADD        SP2                        IF U1,2<V1,2
46        0057  37116777575      FAS4  SP0       INC        RSP1 ROD              READ V3, SP1_V ABS ADDR+2
47        0060  26337777777            OPND ADD        SP0                      SAVE U3
48        0061  30763127632            RB   RD   XOR             CLO  POS       CLO
49        0062  37777777417                 ADD                 SF2            SF2 IF USGN<>VSGN
50        0063  01176777577            SP1  INC        BUS  ROD                 READ V4
51        0064  26306340067            OPND JMP  FAS5 SP1            F3        SAVE V3; U1,2=V1,2?
52        0065  16767417155            SP0  UBUS SUB             CTE  NZRO       YES, SF1 IF
53        0066  26767777153            RA   OPND SUB             CTE               U3,4>=V3,4
54        0067  23176777757      FAS5  SM   INC        BUS  WRS
55        0070  31177777437            RC   ADD        BUS  DATA               SAVE W ABS ADDR AT SM+1
```

```
  56          0071  26626140074          OPND JMP  EAS6 RC         F1         SAVE U4
  57          0072  37312377255     SP0        PEPC       SP1   INCN       EXCH U1,3,4 AND
  58          0073  01337767257          SP1   ADD        SP0   INCN UNC    V1,3,4 IF U<V
  59                                *
  60                                *    UNPACK, V1F-4_ -V1F-4 IF F2, AND ALIGN V WITH U WITH BIT7;
  61                                *      RB,SP2,SP0,RA=U, RD,SP3,SP1,RC=V, F2 IF SUB.
  62                                *    EXIT TO ZANS IF ABS(V)=0 OR UEXP-VEXP>56: F1,OPND_MSGN,EXP,
  63                                *      RB,SP3,1,(FORCED)RBUS_U1F-4, RD,SP2,0,RC_V1F-4, F3_0, CTR MOD.
  64                                *
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
  65          0074  32357777317  EAS6        RB   ADD        CTRL 4PF           CTR_U1F, F1_USGN=MSGN
  66          0075  30641677700               RD   FOMN       FB   077700        RB_VEXP
  67          0076  32163777625          RBUS RB   AND        RBS  OPND          OPND_UEXP=WEXP
  68          0077  30601600077               RD   FOMN       FD   000077        RD_V1F
  69          0100  31773017170          RD   RC   IOR                CF3  NZRO  CF3; CK V2,3=0 IF V1,4=0
  70          0101  14642360222               CTRL JSR  ZP23 PB         UNC        (RB_U1F, RET WITH RB_0)
  71          0102  30611560100               RD   ROM        RD   0100 F2        RD_V1F INCL LEADING ONE,
  72          0103  37766360113                    JMP  EAS7                UNC      JMP IF ADD
  73          0104  31627507777               RC   SUB        RC         CRRY      ELSE V_ -V: V4_ -V4, ZERO?
  74          0105  01316407777               SP1  INC        SP1        ZERO        NO: V3_ -V3-1, STM V2,1
  75          0106  16307507777               UBUS SUB        SP1        CRRY        V3_ -V3 IF V4=0, ZERO?
  76          0107  25536407777               SP3  INC        SP3        ZERO        NO; V2_ -V2-1, STM V1
  77          0110  16527507777               UBUS SUB        SP3        CRRY        V2_ -V2 IF V3,4=0, ZERO?
  78          0111  30776777777               RD   INC                               NO: V1_ -V1-1
  79          0112  16607777777               UBUS SUB        RD                     V1_ -V1 IF V2,3,4=0
  80          0113  14651600100  EAS7         CTRL ROM        FB   000100        RB_U1F INCL LEADING ONE
  81          0114  26767407772          RB   OPND SUB                ZERO       CTR_VEXP-UEXP+1;
  82          0115  16371410100               UBUS ROM        CTRH 0100 NZRO        IF UEXP-VEXP<2 ADJ
  83          0116  17366360124               SBUS JMP  EAS8 CTRH       UNC        V LEFT, CTR_VEXP-UEXP
  84          0117  16771607000               UBUS ROM                  007000
  85          0120  16766130225               UBUS JMP  UAN1                NEG  W_U IF UEXP-VEXP>56
  86          0121  30772337277               RD   PEPC                 INCT CTRM  ADJ V RIGHT
  87          0122  31600733276          UBUS RC   OASP SP1  RD   INCT CTRM        UEXP-VEXP-2 BITS
  88          0123  17626360130               SBUS JMP  EAS9 RC         UNC
  89                                *
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING ( 9)  ***  UBUS ON SBUS OR SL1 MISSING FROM OASL (TASL ON /20)
  90          0124  30600372771  EAS8 RC   RD   OASL SL1  RD
  91          0125  37637737777               ADD        RC         CTRM       V LEFT 1 IF CTRM
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING ( 9)  ***  UBUS ON SBUS OR SL1 MISSING FROM OASL (TASL ON /20)
  92          0126  30600372776          UBUS RD   OASL SL1  RD                    ELSE V LEFT 2
  93          0127  37637777771          RC        ADD       RC
  94                                *
  95          0130  35537777777  EAS9        SP2  ADD        SP3
  96          0131  25737777777               SP3  ADD        SP2
  97          0132  37317777775          SP0       ADD        SP1                  RB,SP3,SP1,RA_U1F-4
  98          0133  01337777777               SP1  ADD        SP0                  RD,SP2,SP0,RC_V1F-4
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING ( 9)  ***  UBUS ON SBUS OR SL1 MISSING FROM OASL (TASL ON /20)
  99          0134  32760372773          RA   RB   OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 100          0135  16640372777               UBUS OASL SL1  RB                   ALIGN U WITH BIT7
 101                                *
 102                                *    U_U+V, EXIT TO ZANS IF ANSW=ZERO;
```

```
103                                *       RB,SP3,SP1,(FORCED)RBUS=U1F-4, RD,SP2,SP0,RC=V1F-4, OPND=WEXP.
104                                *     RD_WEXP-2, RB,SP3,SP1,RA_W1F-4*4, F3_1.
105                                *
106           0136  31677517777                PC    ADD        FA          NCRY      U4_U4+V4, CRRY?
107           0137  37776407775          SP0         INC                    ZERO      YES; V3=V3+1, CRRY?
108           0140  16317517774          SP1   UBUS  ADD        SP1         NCRY      NO; U3_U3+V3, CRRY?
109           0141  35776407777                SP2   INC                    ZERO      YES; V2=V2+1, CRRY?
110           0142  25537517776          UBUS  SP3   ADD        SP3         NCRY      NO; U2_U2+V2, CRRY?
111           0143  30776777777                RD    INC                              YES; V1F=V1F+1
112           0144  16657417072          RB    UBUS  ADD        RB    SP3   NZRO      U1F_U1F+V1F, SP3; CHECK
113           0145  33342000222                RA    JSB   ZP23  CTRL        ZERO        W2,3=0 IF W1F,4=0, CTR_0
114           0146  26611777600                OPND  POM         RD    177600          RD_WEXP-2 (U=W*4)
115           0147  23176777777                SM    INC         BUS   ROS             READ W ABS ADDR FROM SM+1
116                                *
117                                *     NORMAL EXIT FOR FADD/FSUB AND FMPY/FDIV (NOR3).
118                                *     NORMALIZE AND ROUND W: RB,SP3,SP1,RA=W1F-4, F3,CTR=0 IF NOP3.
119                                *     CTR_SHIFT CNT, F2_1 IF SHIFT RIGHT ELSE F2_0, IF NORM,2 F3_0.
120                                *
121           0150  32341417400    NORM        RB    ROMN        CTRL  7400  NZRO      CTR_0,
122           0151  37766360162                      JMP   NOR6                UNC       ADJ LEFT IF W(0:7)=0
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SP1 MISSING FROM OASR (TASR ON /20)
123           0152  33640773272    NOR2  RB    RA    OASR  SP1   RB    INCT      ELSE ADJ RIGHT, SP1
124           0153  37677777177                ADD         RA    CF3             CF3
125           0154  32761407400    NOR3        RB    ROMN              7400  ZERO      SP1 IF W(0:7)<>0
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SP1 MISSING FROM OASR (TASR ON /20)
126           0155  33640773272          RB    RA    OASR  SP1   RB    INCT
127           0156  33676517417                RA    INC         RA    SF2   NCRY      F2_1, RND
128           0157  01316507777                SP1   INC         SP1         CRRY
129           0160  37766360171                      JMP   PACK                UNC       PACK OR
130           0161  37766360167                      JMP   NOF7                UNC         FINISH RND IF NEC
131                                *
132           0162  32772067177    NOR6        RB    REPC              CF3   BIT8      CF3
133           0163  16640062273          FA    UBUS  OASL  SP1   RB    INCT  BIT8      NORM LEFT TO BIT8 IF NEC
134           0164  37676517425          RBUS        INC         FA    CF2   NCRY      F2_0: RND
135           0165  01316507777                SP1   INC         SP1         CRRY
136           0166  37766360171                      JMP   PACK                UNC       PACK OR
137           0167  25536517777    NOR7        SP3   INC         SP3         NCRY        FINISH RND IF NEC
138           0170  32656777777                RB    INC         RB
139                                *
140                                *     PACK AND STORE W AT (OPND), TEST FOR UN/OVFL;
141                                *     F1=WSGN, IF F3 RIGHT SHIFT INHIBITED AND ABS(W)=0 ALLOWED,
142                                *     RD=WEXP, RB,SP3,SP1,RA=W1F-4, CTR=DELTA EXP FROM
143                                *     NORMALIZATION (F2 IF NORMALIZED RIGHT), OVFL CLR, SR=3: SR_0.
144                                *
145           0171  26137747541    PACK SR   OPND  ADD         BSP0  WRD   F3        SP0_W ABS ADDR+3
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
146           0172  33640773772          RB    RA    OASR  SR1   RB                SR1 AFTER RND IF NF3
147           0173  33177777437                RA    ADD         BUS   DATA        STORE W4
148           0174  01633157716          UBUS  SP1   IOR         RC    CCG   NF1   RC_W4 IOR W3; CCG
149           0175  37777777677                      ADD               CCL           IF W POS ELSE CCL
150           0176  37127377555          SP0         CAD         BSP0  WRD
151           0177  01177567437                SP1   ADD         BUS   DATA  F2    STORE W3
```

```
152          0200  15767777770          RD   CTRH SUB                                  EXP=EXP+-NORM SHIFT CNT
153          0201  32657777336          UBUS RB   ADD        PB   FHB                  PACK SGN AND EXP INTO W1
154          0202  17735372305          RBUS SBUS CRS  SF1    SP2  HBF                  SP2_CSL(ABS(W1)), SF1 IF
155          0203  37127377555          SP0       CAD        BSP0 WRD                                  POSS UNFL
156          0204  25177777437               SP3  ADD        BUS  DATA                 STORE W2
157          0205  25633377051          RC   SP3  IOR        RC   CLSR                 RC_W4 IOR W3 IOR W2, SR_0
158          0206  37127377555          SP0       CAD        BSP0 WRD                  SP0_W DB REL ADDR
159          0207  32177777437               RB   ADD        BUS  DATA                 STORE W1
160          0210  35766030214               SP2  JMP  EFOV                  ODD       JMP  IF UN(F1)/OVFL
161          0211  35773007471          RC   SP2  IOR                  SF1  ZERO      POSS UNFL IF ABS(W)=0
162          0212  37777757777               ADD                             NEXT      ELSE DONE
163          0213  32726340740               RB   JMP  ECP5 SP2              F3        JMP, CCA IF ABS(W)=0 AND
164                                *                                                       ABS(W)=0 IS OK (2C JMP)
165                                *
166                                *    ADD/SUB EXP OVFL: 1000              UNFL: 0000 (W=0) TO 1711
167                                *    MPY               1000 TO 1377          0000 (W=0) TO 1400
168                                *    DIV               1000 TO 1377          0000 (W=0) TO 1400
169                                *
170                                *    OVFL: F1=0, PARAM_%10  UNFL: F1=1, PARAM_%11  EDZR(EFV1):
171                                *      UBUS=STA, F1=0, CTR=2, PARAM_%12; SP0=W ABS ADDR, SR<4.
172                                *    EXIT TO TRP0 WITH SP3_PARAM AND IF TRAPS ENABLED TOS_W DB REL ADDR.
173                                *      TRP0 SETS UP INTERRUPT 1,25 IF TRAPS ENABLED ELSE SOV,NEXT.
174                                *
175          0214  24357771777    EFOV      STA  ADD  LIZ  CTRL                       CTR_0
176          0215  16777522776    EFV1 UBUS UBUS ADD  SF1                 POS       STA(2)=1? (TRAPS ENABLED)
177          0216  22207777775         SP0  DB   SUB       PUSH                      YES, TOS_W DB REL ADDR
178          0217  14531550010              CTRL MOM       SP3  0010 NF1            PARAM_%10+CTR
179          0220  16536777777              UBUS INC       SP3                       +1 IF F1
180          0221  37571603134                   MOM       PAR  TRP0                 EXIT TO TRP0
181                                *
182                                *    ZR23: RET WITH RB_0 IF SP1,SP3<>0; ELSE ZAN2 WITH RD_0 OR UAN1.
183                                *    ZAN1: F3=1, SR=3, OVFL CLR; EXIT TO ZAN2 WITH RD_0.
184                                *    UAN1: F1,OPND=WSGN,EXP, RB,SP3,SP1 (SP2,SP0 IF NF3),RA=W1[F1-4,
185                                *          SR=3, OVFL CLR; EXIT UAN2 WITH F1,RD,RB,SP3,SP1,RA_W, F3_1.
186                                *    ZAN2: RD=0, SR=3, OVFL CLR; UAN2 WITH F1,RB,SP3,SP1,RA_0, F3_1.
187                                *    UAN2: F1,RD,RB,SP3,SP1,RA=W, F3=1, SR=3, OVFL CLR;
188                                *          EXIT TO PACK WITH OPND_W ABS ADDR, CTR_0.
189                                *
190          0222  25773007774    ZR23 SP1  SP3  IOR                       ZERO
191          0223  37657707777              ADD       RB                  RSR       RET, RB_0 IF SP1,SP3<>0
192          0224  37606340231    ZAN1      JMP  ZAN2 RD                  F3        RD_0, ZAN2 IF F3
193          0225  32641600077    UAN1 RB   ROMM      RB   000077                  ELSE UAN1; RB_W1F
194          0226  26606340235         OPND JMP  UAN2 RD                  F3        RD_WEXP, JMP IF SP3,SP1 OK
195          0227  37317777075    SP0       ADD       SP1  SF3                     SP1_SP0=W3, SF3
196          0230  35526360235              SP2  JMP  UAN2 SP3             UNC       SP3_SP2=W2
197                                *
198          0231  37657777457    ZAN2      ADD       PB   CF1                     PB_0, CF1
199          0232  37537777077               ADD       SP3  SF3                     SP3_0, SF3
200          0233  37317777777               ADD       SP1                          SP1_0
201          0234  37677777777               ADD       RA                           RA_0
202                                *
203          0235  23176777777    UAN2      SM   INC       BUS  POS                 READ W ABS ADDR FROM SM+1
204          0236  37346360171               JMP  PACK CTRL            UNC         CTR_0; PACK AND STORE W
205                                *
206                                *
```

```
207                          *         EMPY/EDIV
208                          *    EDIV: EXIT TO EDIV WITH CONDITIONS LISTED AT EDIV.
209                          *    EMPY: EMPTY RD, SAVE W ABS ADDR AND X AT SM+1,3, FETCH U AND V,
210                          *    CK BOUNDS OF U,V,W, CLO, CK FOR U OR V ZERO, CALC W FROM PARTIAL
211                          *    PRODUCTS P4-16 SHOWN BELOW.  IN GENERAL, (PORTIONS OF) SUM OF
212                          *    PREVIOUS PARTIAL PRODUCTS ADDED INTO MPAD CALCULATING THE NEXT
213                          *    PARTIAL PRODUCT.  CRRY IS TESTED AS INDICATED.
214                          *
215                          *                                          MSP4  MSP4 LSP4 LSP4  U2*V4
216                          *                                        C MSP5  MSP5 LSP5 LSP5  V3*U3
217                          *                                      CC MSP6  MSP6 LSP6 LSP6  U4*V2
218                          *                                    MSP7 LSP7  LSP7            U1*V4
219                          *                                    MSP8 LSP8  LSP8            V1*U4
220                          *                              C  MSP9  MSP9 LSP9  LSP9         U3*V2
221                          *                            CC MSP10 MSP10 LSP10 LSP10         V3*U2
222                          *                       MSP11 LSP11 LSP11                       V1*U3
223                          *                       MSP12 LSP12 LSP12                       U1*V3
224                          *                 C MSP13 MSP13 LSP13 LSP13                     U2*V2
225                          *           MSP14 LSP14 LSP14                                   V1*U2
226                          *           MSP15 LSP15 LSP15                                   U1*V2
227                          *     MSP16 LSP16                                               U1*V1
228                          *        W1    W1    W2    W2    W3    W3    W4    W4
229                          *
230                          &0243 (10243)
231          0243 22176777572 EMPY PB   DB   INC        BUS  ROD            READ U2
232          0244 22317777773      PA   DB   ADD        SP1                 SP1_V ABS ADDR
233          0245 23662360770           SM   JSB  PSHM PA        UNC       PA_SM, EMPTY RD
234          0246 01766777773      PA   SP1  RNDT                          SM>=V?
235          0247 01137777561      SR   SP1  ADD        BSP0 ROD            READ V4
236          0250 26602360030           OPND JSB  EAS1 RD        UNC       SAVE U2; CK V>=DL,SM>=W
237          0251 22317777772      RB   DB   ADD        SP1                 SP1_U ABS ADDR
238          0252 37127377575      SP0       CAD        BSP0 ROD            READ V3
239          0253 26642360035           OPND JSB  EAS2 RB        UNC       SAVE V4; CK SM>=U, U,W>=DL
240          0254 37127377575      SP0       CAD        BSP0 ROD            READ V2
241          0255 26737777237           OPND ADD        SP2  OCSB           SAVE V3, SR_2
242          0256 00766030426           CIR  JMP  EDIV             ODD       JMP IF EDIV
243          0257 23176777741      SR   SM   INC        BUS  WRS
244          0260 37177777426      X         ADD        BUS  DATA           SAVE X AT SM+3
245          0261 30537777637           RD   ADD        SP3  CLO            LOAD U2 FOR P4, CLO
246          0262 26677777777           OPND ADD        RA                  SAVE V2
247          0263 01177777561      SR   SP1  ADD        BUS  ROD            READ U3
248          0264 37772607777           REPN                    20
249          0265 16774333272      RB   UBUS MPAD SR1        INCT CTRM      P4=U2*V4
250          0266 17557777777           SBUS ADD        X                     X,SP3_P4
251          0267 23176777757           SM   INC        BUS  WRS
252          0270 31177777437           RC   ADD        BUS  DATA           SAVE W ABS ADDR AT SM+1
253          0271 26637777777           OPND ADD        RC                  SAVE U3
254          0272 01176777561      SR   SP1  INC        BUS  ROD            READ U4
255          0273 35537777777           SP2  ADD        SP3                 LOAD V3 FOR P5
256          0274 25772607777           SP3  REPN                    20
257          0275 16774333271      RC   UBUS MPAD SP1        INCT CTRM
258          0276 17557517766      X    SBUS ADD        X         NCRY      P5=(V3*U3+LSP4)+MSP4
259          0277 37156777017           INC        SBR  ABS                    ABS=BNK,X,SP3_P5
260          0300 26317777777           OPND ADD        SP1                 SAVE U4
261          0301 01177777577           SP1  ADD        BUS  ROD            READ U1
```

```
ADDRESS   CONTENTS      LABL PBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

0302  01537777077            SP1  ADD        SP3  SF3              LOAD U4 FOR P6, SF3
0303  25772607217            SP3  REPN            INSR 20          SR_3
0304  16774333273       RA   UBUS MPAD SP1        INCT CTRM
0305  17557517766       Y    SBUS ADD        Y         NCRY        P6=(U4*V2+LSP5)+MSP5
0306  03156777017            RBR  INC        SBR  APS                ABS-BNK,X,SP3_P6
0307  37167377575       SP0       CAD        BUS  ROD              READ V1
0310  26357777317            OPND ADD        CTRL HBF              CTR_VIF, F1_USGN
0311  16326010314            UBUS JMP  EMP2 SP0       NZRO        SAVE U1, JMP IF NZRO
0312  31773017770       RD   RC   IOR                 NZRO
0313  01766000020            SP1  JMP  ZAN3            ZERO        JMP IF U=0
                       *
                       *    SP0,RD,RC,SP1=U, CTR=U1F, OPND,RA,SP2,RB=V, ABS-BNK,X,SP3=P6,
                       *    F1=USGN, F3=1, SR=3
                       *
0314  14531600100      EMP2      CTRL POM        SP3  000100         LOAD U1F INCL LEADING ONE
0315  37772707766       X        REPN                 10
0316  16774333272       RB   UBUS MPAD SP1        INCT CTRM        P7=U1*V4+MSP6
0317  17557771777            SBUS ADD  LLZ  X                        (SP=P6 CRRY)
0320  17657774777            SBUS ADD  RRZ  RB                       LH X,RH RB,LH SP3_P7
0321  35773377772       RB   SP2  IOR
0322  33773017776       UBUS RA   IOR                 NZRO
0323  26766000020            OPND JMP  ZAN3            ZERO        JMP IF V=0
0324  26357777337            OPND ADD        CTRL FHB              CTR_VIF
0325  16761777700            UBUS POEN       177700                VEXP WITH USGN
0326  16177777635       SP0  UBUS ADD        BUS  OPND             OPND_U1+VEXP
0327  26763377315       SP0  OPND XOR             HBF              F1_WSGN
0330  14331600100            CTRL POM        SP0  000100         SP0_VIF INCL LEADING ONE
0331  16537777777            UBUS ADD        SP3                   LOAD VIF INCL LEADING ONE
0332  25772706172       RB   SP3  REPN SWAP       CF3  10          CF3
0333  16774333274       SP1  UBUS MPAD SP1        INCT CTRM
0334  17657774777            SBUS ADD  RRZ  RB                     P8=((V1*U4+LSP7)+MSP7)
0335  17777770766       X    SBUS ADD  LPZ                            +P6 CRRY
0336  03557777016       UBUS RBR  ADD        Y    ABS                X,RH RB,LH SP3_P8
                       *
                       *    OPND,RD,RC=UEXP+VEXP,UIF-3, SP0,RA,SP2=VIF-3, SP1 EMPTY,
                       *    X,RH RB,LH SP3=P8, F1=WSGN, F3=0, SR=3, ABS-BNK MODIFIED
                       *
0337  31537777777            RC   ADD        SP3                   LOAD U3 FOR P9
0340  25772606772       RB   SP3  REPN SWAP            20
0341  16774333273       RA   UBUS MPAD SP1        INCT CTRM
0342  17557517046       X    SBUS ADD        Y    CLSR NCRY        P9=(U3*V2+LSP9)+MSP8
0343  37777777217                 ADD             INSR               SR,X,SP3_P9
0344  35537777777            SP2  ADD        SP3                   LOAD V3 FOR P10
0345  25772607777            SP3  REPN                20
0346  16774333270       RD   UBUS MPAD SP1        INCT CTRM
0347  17557517766       X    SBUS ADD        Y         NCRY        P10=(V3*U2+LSP9)+MSP9
0350  37777777217                 ADD             INSR               SR,X,SP3_P10
0351  25657777777            SP3  ADD        RB                    RB_LSP10=W4
                       *
                       *    OPND,RD,RC=UEXP+VEXP,UIF-3, SP0,RA,SP2=VIF-3,
                       *    SR,X,SP3=P10, F1=WSGN, RB=W4, F3=0, ABS-BNK UNDF, SP1 EMPTY
                       *
0352  37537777775       SP0       ADD        SP3                   LOAD VIF FOR P11
0353  37772707766       X         REPN                 10
0354  16774333271       RC   UBUS MPAD SP1        INCT CTRM        P11=V1*U3+MSP10
```

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

0355   17557771777            SBUS ADD  LLZ  X                       (SR=P10 CRRY)
0356   17317774777            SBUS ADD  RPZ  SP1                     LH X,RH SP1,LH SP3_P11
0357   35637777777            SP2  ADD       PC                      PC_V3
0360   26721600077            OPND ROMN      SP2  000077            SP2_UIF W/O LEADING ONE
0361   16531600100            UBUS ROM       SP3  000100            LOAD UIF INCL LEADING ONE
0362   25772706774       SP1  SP3  REPN SWAP          10
0363   16774333271       PC   UBUS MPAD SR1       INCT CTRM
0364   17317774777            SBUS ADD  RPZ  SP1                     P12=((U1*V3+LSP11)+MSP11)
0365   17777770766       X    SBUS ADD  LPZ                           +P10 CRRY
0366   16557777761       SR   UBUS ADD       X                       X,RH SP1,LH SP3_P12
0367   30537777777            RD   ADD       SP3                     LOAD U2 FOR P13
0370   25772606774       SP1  SP3  REPN SWAP          20
0371   16774333273       RA   UBUS MPAD SP1       INCT CTRM
0372   17557517046       X    SBUS ADD       X    CLSP NCRY          P13=(U2*V2+LSP12)+MSP12
0373   37777777217            ADD            INSR                    SR,X,SP3_P13
0374   25317777777            SP3  ADD       SP1                     SP1_LSP13=W3
                         *
                         *    OPND=UEXP+VEXP,U1F, SP2,RD=U1F-2 W/O LEADING ONE, SP0,RA=V1F-2,
                         *    SR,X,SP3=P13, F1=WSGN, SP1,RB=W3-4, F3=0, ABS-BNK UNDF, PC EMPTY
                         *
0375   37537777775       SP0       ADD       SP3                     LOAD VIF FOR P14
0376   37772707766       X         REPN              10
0377   16774333270       RD   UBUS MPAD SP1       INCT CTRM          P14=V1*U2+MSP13
0400   17617771777            SBUS ADD  LLZ  RD                      (SR=P13 CRRY)
0401   17637774777            SBUS ADD  RPZ  PC                      LH RD,RH RC,LH SP3_P14
0402   35531600100            SP2  ROM       SP3  000100            LOAD UIF INCL LEADING ONE
0403   25772706771       RC   SP3  REPN SWAP          10
0404   16774333273       RA   UBUS MPAD SP1       INCT CTRM
0405   17637774777            SBUS ADD  RPZ  PC                      P15=((U1*V2+LSP14)+MSP14)
0406   17777770770       RD   SBUS ADD  LPZ                           +P13 CRRY
0407   16617777041       SR   UBUS ADD       RD   CLSR                RD,RH RC,LH SP3_P15
0410   25637776211       RC   SP3  ADD  SWAP RC   INSR                RC_RH RC,LH SP3=W2, SR_1
0411   35531600100            SP2  ROM       SP3  000100            LOAD UIF INCL LEADING ONE
0412   30772707217            RD   REPN            INSP 10          SR_2
0413   16774333275       SP0  UBUS MPAD SP1       INCT CTRM          P16=U1*V1+MSP15
0414   17677777777            SBUS ADD       RA                      RH RA,LH SP3_P16
                         *
                         *    OPND=UEXP+VEXP,U1F, F1=WSGN, RA(10:15),LH SP3,PC,SP1,RB=W,
                         *    F3=0, SR=2, ABS-BNK UNDF, SP0,SP2,RD,X EMPTY
                         *
                         *    RESTORE X, READ W ABS ADDR, SR_3, CTR_0,
                         *      RD_WEXP-1, RB,SP3,SP1,RA_W1F-4*2.
                         *    EXIT TO NOR3 TO FINISH NORMALIZATION, RND AND PACK.
                         *
0415   23176777761       SR   SM   INC       BUS  ROS                 READ X FROM SM+3
0416   26761777700            OPND ROMN           177700
0417   16611737700            UBUS ROM       RD   137700            RD_UFXP+VEXP-257=WFXP-1
0420   31537777777            RC   ADD       SP3                     LOAD W2 FOR NORMALIZATION
0421   25772736773       RA   SP3  REPN SWAP          05
0422   32640733276       UBUS RB   OASR SP1  RB   INCT CTRM          RB(7:15),SP3,SP1,RA_W1F-4*2
0423   17677777217            SBUS ADD       RA   INSR                SR_3, CTR_0
0424   23176777777            SM   INC       BUS  ROS                 READ W ABS ADDR FROM SM+1
0425   26546360154            OPND JMP  NOR3 X         UNC            RESTORE X; NORM,RND,PACK
                         *
                         *
```

```
372                                 *          EDIV
373                                 *     ENTER FROM EMPY WITH U,V,W BOUNDS CHECKED AND SP1=U ABS ADDR,
374                                 *       SP0=V2 ABS ADDR, RC=W ABS ADDR, RD=U2, OPND,SP2,RB=V2-4, SR=2.
375                                 *     FINISH FETCHING U,V; SAVE W ABS ADDR, WEXP, X AT SM+1; CLO; CK U OR
376                                 *       V=0; SHIFT V LEFT 9 INSERTING LEADING 1, U LEFT 8; SF3 IF W POS;
377                                 *       CTR,SP3,SP1,RC,RD_U, RA,SP2,OPND,RB_V, F3_ -WSGN, F1 MODIFIED.
378                                 *
379          0426  37167377575      EDIV SP0       CAD       BUS  ROD            READ V1
380          0427  26537777277           OPND ADD       SP3  INCT             SAVE V2 (TEMP IN SP3)
381          0430  23176777757           SM   INC       BUS  WPS
382          0431  31177777437           RC   ADD       BUS  DATA             SAVE W ABS ADDR AT SM+1
383          0432  35773377272      RB   SP2  IOR            INCT             CTR_2
384          0433  25773017636      UBUS SP3  IOR            CLO  NZRO       CLO
385          0434  26766000661           OPND JMP  EDZR            ZERO       EDZR IF V=0
386          0435  26621777700           OPND ROMN      RC   177700           RC_VSGN,EXP
387          0436  23176777741      SR   SM   INC       BUS  WPS
388          0437  37177777426      X         ADD       BUS  DATA             SAVE X AT SM+3
389          0440  35317777477           SP2  ADD       SP1  SF1              LOAD V3 FOR SHIFT, SF1
390          0441  01137777577           SP1  ADD       RSP0 ROD              READ U1
391          0442  26772677337           OPND REPN           FHB  11
392          0443  16660332272      RB   UBUS OASI SI1 RA   INCT CTRM       SHIFT V LEFT 9, INSERTING
393          0444  37657777265      RBUS      ADD       RB   INCT               LEADING ONE; CTR_1
394          0445  26343377311      RC   OPND XOR       CTRL HBF              CTR_UIF, SF1 IF VSGN<>USGN
395          0446  14136777575      SP0  CTRL INC       RSP0 ROD              READ U3
396          0447  26541677700           OPND ROMN      X    077700           X_UEXP
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
397          0450  31723777765      RBUS RC   AND       SP2                  SP2_VEXP
398          0451  23177777741      SR   SM   ADD       BUS  WRS              SAVE WEXP-256
399          0452  35167777426      X    SP2  SUB       BUS  DATA               AT SM+2 (UEXP-VEXP)
400          0453  25737777777           SP3  ADD       SP2                  SAVE V2
401          0454  30537771777           RD   ADD  LIZ  SP3                  SP3_LH U2
402          0455  30637775777           RD   ADD  RLZ  RC                   RC_RH U2
403          0456  37176777575      SP0       INC       BUS  ROD            READ U4
404          0457  26337775777           OPND ADD  RIZ  SP0                  SP0_RH U3
405          0460  17617770217           SBUS ADD  LPZ  RD   INSR           RD_LH U3, SR_3
406          0461  17553147770      RD   SBUS IOR       Y         F1         X_U2 IOR U3,
407          0462  14777777066      X    CTRL ADD            SF3               SF3 (W POS) IF USGN=VSGN
408          0463  16773017766      X    UBUS IOR                 NZRO       UEXP,UIF IOR U2 IOR U3
409          0464  26606000020           OPND JMP  ZAN3 RD        ZERO       RD_0, JMP IF ABS(U)=ZERO
410          0465  30317777231      RC   RD   ADD       SP1  DCSR             SAVE U2 (SL8), SR_2
411          0466  01177777637           SP1  ADD       BUS  OPND             SAVE V3
412          0467  26617775777           OPND ADD  RIZ  RD                   SAVE U4 (SL8)
413          0470  17777770777           SBUS ADD  LPZ
414          0471  16637777775      SP0  UBUS ADD       RC                   SAVE U3 (SL8)
415                                 *
416                                 *     U=CTR,SP3,SP1,PC,RD   01 14X  16X  16X  7X X 80
417                                 *     V=RA,SP2,OPND,RB       1X 14X  16X  16X  7X 0 80
418                                 *
419                                 *     CALC: Q1=U1,2/V1 CARRIED OUT 17 PLACES (16 SIGNFICANT BITS).
420                                 *           R1=R11,U3,4-Q1*V2,3,4, IF R1<0 R1_R1+V+[V], Q1_Q1-1-[1].
421                                 *
422                                 *     Q1=X                0 14X X
423                                 *                         1 14X X
424                                 *     R1=SP3,SP1,RD,SP0          X 15X  16X  8X 80
425                                 *     Q1*V2,3,4                  X 15X  16X  8X 8X  7X 90
```

```
ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

426                    *    V IF ADD BACK            1 15X  16X  8X 8X  7X 90
427                    *
428                    *    ON COMPLETION DONE WITH V4, RB,RC EMPTY, F2_0, F1 MODIFIED.
429                    *
430       0472  14771600100 ED10       CTRL ROM              000100
431       0473  25772576776      UBUS SP3  REPN SWAB            21        U1 (SL8) WITH LEADING ONE
432       0474  33764332276      UBUS RA   DVSB SL1      INCT CTRM       SP1_Q1=U1,2/V1
433       0475  37557573425      RBUS      ADD  SP1  X    CF2  NF2        CF2
434       0476  16551700000           UBUS ROM        X    100000        X_R11
435       0477  32537770777           RB   ADD  LPZ  SP3                LOAD V4 FOR P13
436       0500  37772707777                REPN                10
437       0501  16774333274      SP1  UBUS MPAD SP1        INCT CTRM       P13=V4*Q1
438       0502  17537777777           SBUS ADD       SP3                    SP3_MSP13
439       0503  25327777157           SP3  SUB       SP0  CTF               SP0_ -LSP13=R14 (F1 IF 0)
440       0504  26537777777           OPND ADD       SP3                LOAD V3 FOR P12
441       0505  25772607777           SP3  REPN                20
442       0506  16774333274      SP1  UBUS MPAD SP1        INCT CTRM       P12=V3*Q1+MSP13
443       0507  17537777777           SBUS ADD       SP3                    SP3_MSP12
444       0510  25607157150      RD   SP3  CAD       RD   CTF  NF1          RD_U4-LSP12-(1 IF NF1)
445       0511  17607777150      RD   SBUS SUB       RD   CTF                 =R13 (F1 IF "POS")
446       0512  35537777777           SP2  ADD       SP3                LOAD V2 FOR P11
447       0513  25772607777           SP3  REPN                20
448       0514  16774333274      SP1  UBUS MPAD SP1        INCT CTRM       P11=V2*Q1+MSP12
449       0515  17537777777           SBUS ADD       SP3                    SP3_MSP11
450       0516  25627157151      RC   SP3  CAD       RC   CTF  NF1          RC_U3-LSP11-(1 IF NF1)
451       0517  17627777151      RC   SBUS SUB       RC   CTF                 =R12 (F1 IF "POS")
452       0520  25527157146      X    SP3  CAD       SP3  CTF  NF1          SP3_R11-MSP11-(1 IF NF1)
453       0521  25527777146      X    SP3  SUB       SP3  CTF                 =R11 (F1 IF POS)
454       0522  01557557777           SP1  ADD       X         NF1        X_Q1,
455       0523  31306360535           RC   JMP  ED20 SP1       UNC          JMP, SP1_R12 IF R1 POS
456       0524  32337517275 ED12 SP0  RB   ADD       SP0  INCT NCRY       R14_R14+V4, CRRY?
457       0525  26776407777           OPND INC                 ZERO       YES; V3=V3+1, CRRY?
458       0526  16617517770      RD   UBUS ADD       RD        NCRY       NO; R13_R13+V3, CRRY?
459       0527  35776407777           SP2  INC                 ZERO       YES; V2=V2+1, CRRY?
460       0530  16637517771      RC   UBUS ADD       RC        NCRY       NO: R12_R12+V2, CRRY?
461       0531  25536767153      RA   SP3  INC       SP3  CTF  UNC        YES; R11_R11+V1+1,F1 IF POS
462       0532  25537777153      RA   SP3  ADD       SP3  CTF            NO; R11_R11+V1, SF1 IF POS
463       0533  31306150524           RC   JMP  ED12 SP1       NF1        ADD BK AGAIN IF R1 NEG,
464       0534  14547777766      X    CTRL SUB       X                     SP1_R12, Q1_Q1-CTR
465                    *
466                    *    Q1=X                15X X
467                    *    R1=SP3,SP1,RD,SP0      X 15X  16X  16X  7X 90
468                    *    V=RA,SP2,OPND          1 15X  16X  16X
469                    *
470                    *    CALC: Q2=R11,12/V1 CARRIED OUT 17 PLACES (17 SIGNIFICANT BITS).
471                    *          IF Q2<2**16: R2=R21,13,14-Q2*V2,3.
472                    *          IF Q2>=2**16: R2=R21,13,14-2**16*V2,3; IF R2>=0 Q2,3,4_ -1;
473                    *            IF Q2>2**16 THEN R2_R2+V1,2,3-V2,3, Q2_Q2-1,
474                    *            IF R2>=0 Q2,3,4_ -1.
475                    *          IF R2<0: R2_R2+V+[1], Q2_Q2-1-[1].
476                    *
477                    *    Q2=RB               (X)  15X X
478                    *    R2=SP3,SP1,SP0             X 15X  16X  7X 90
479                    *    (Q2<2**16+1)*V2,3          X 15X  16X  7X 9X
480                    *                  V2,3                  16X  7X 9X
```

```
ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

                       *    V1,2,3 IF ADD BK            1 15X  16X  7X 9X
                       *
                       *    ON COMPLETION DONE WITH V3, RC,RD,OPND EMPTY, F1,2 MODIFIED.
                       *
0535  25772577777      ED20      SP3  REPN                    21
0536  33764332276           UBUS RA   DVSB SL1         INCT CTRM       SP1_Q2=R11,12/V1
0537  37655123765           RBUS      CRS  SP1  RB            POS       F2,RB(1:15)_R21,
0540  37766360634                     JMP  ED26               UNC         JMP IF Q2>=2**16
0541  26537577777                OPND ADD       SP3           NF2       LOAD V3 FOR P22,
0542  32651700000                RB   PDM       RB    100000              RB_R21
0543  37772607777                     REPN                   20
0544  16774333274           SP1  UBUS MPAD SP1         INCT CTRM       P22=V3*Q2
0545  17537777777                SBUS ADD       SP3                       SP3_MSP22
0546  25327777155           SP0  SP3  SUB       SP0   CTF                 SP0_R14-LSP22=R23
0547  35537777777                SP2  ADD       SP3                     LOAD V2 FOR P21 \(F1="POS")
0550  25772607777                SP3  REPN                   20
0551  16774333274           SP1  UBUS MPAD SP1         INCT CTRM       P21=V2*Q1+MSP22
0552  17537777777                SBUS ADD       SP3                       SP3_MSP21
0553  25627157150           RD   SP3  CAD       RC    CTF  NF1           RC_R13-LSP21-(1 IF NF1)
0554  17627777150           RD   SBUS SUB       RC    CTF                  =R22 (F1 IF "POS")
0555  25527157152           RB   SP3  CAD       SP3   CTF  NF1           SP3_R21-MSP21-(1 IF NF1)
0556  25527777152           RB   SP3  SUB       SP3   CTF                  =R21 (F1 IF POS)
0557  01657557777                SP1  ADD       RB          NF1         RB_Q2,
0560  31306360570                RC   JMP  ED30 SP1         UNC           JMP, SP1_R22 IF R2 POS
0561  26337517275      ED22 SP0  OPND ADD       SP0   INCT NCRY        R23_R23+V3, CRRY?
0562  35776407777                SP2  INC                  ZERO        YES; V2=V2+1, CRRY?
0563  16637517771           RC   UBUS ADD       RC          NCRY        NO: R22_R22+V2, CRRY?
0564  25536767153           RA   SP3  INC       SP3   CTF  UNC         YES; R21_R21+V1+1,F1 IF POS
0565  25537777153           RA   SP3  ADD       SP3   CTF               NO; R21_R21+V1, SF1 IF POS
0566  31306150561                RC   JMP  ED22 SP1         NF1         ADD BK AGAIN IF R2 NEG,
0567  14647777772           RB   CTRL SUB       RB                        SP1_R22, Q2_Q2-CTR
                       *
                       *    Q1,2=X,RB        16X  15X X
                       *    R2=SP3,SP1,SP0             X 15X  16X  16X
                       *    V=RA,SP2                   1 15X  16X
                       *
                       *    CALC: Q3=R21,22/V1 CARRIED OUT 17 PLACES (17 SIGNIFICANT BITS).
                       *          IF Q3<2**16: R3=R31,23-Q3*V2.
                       *          IF Q3>=2**16: R3=R31,23-2**16*V2; IF R3>=0 Q3,4_ -1;
                       *            IF Q3>2**16 THEN R3_R3+V1,2-V2, Q3_Q3-1, IF R3>=0 Q3,4_ -1.
                       *          IF R3<0: R3_R3+V+[V], Q3_Q3-1-[1].
                       *
                       *    Q3=RC            (X)  15X X
                       *    P3=SP0,SP1                 X 15X  16X
                       *    Q3*V2                      X 15X  16X
                       *    V1,2 IF ADD BK             1 15X  16X
                       *
                       *    ON COMPLETION DONE WITH V2, SP2,SP3,RD EMPTY, F1 MODIFIED.
                       *    IF EXIT TO ED40 THEN OPND_WEXP-256, F2_0.
                       *
0570  25772577437      ED30      SP3  REPN                CF2  21        CF2
0571  33764332276           UBUS RA   DVSB SL1         INCT CTRM       SP1_Q3=R21,22/V1
0572  37635123765           RBUS      CRS  SP1  RC            POS       F2,RC(1:15)_R31,
0573  37766360651                     JMP  ED36               UNC         JMP IF Q3>=2**16
0574  35537577777                SP2  ADD       SP3           NF2       LOAD V2 FOR P31,
```

```
      ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

536           0575  31631700000                RC   POM        PC   100000          RC_R31
537           0576  37772607437                     REPN            CF2  20        CF2
538           0577  16774333274           SP1  UBUS MPAD SF1        INCT CTRM      P31=V2*Q3
539           0600  17537777777                SBUS ADD        SP3                   SP3_MSP31
540           0601  25607507775           SP0  SP3  SUB        FD        CRRY        RD_R23-LSP31=R32
541           0602  25327367151           RC   SP3  CAD        SP0  CTF  UNC         SP0_R31-MSP31-[1]
542           0603  25327777151           RC   SP3  SUB        SP0  CTF               =R31 (F1 IF POS)
543           0604  01637557777                SP1  ADD        RC        NF1         RC_Q3,
544           0605  30306360613                RD   JMP  ED34 SP1        UNC           JMP, SP1_R32 IF R3 POS
545           0606  35617517270      ED32 RD   SP2  ADD        RD   INCT NCRY        R32_R32+V2, CRRY?
546           0607  33336767155           SP0  RA   INC        SP0  CTF  UNC         YES; R31_R31+V1+1,F1 IF POS
547           0610  33337777155           SP0  RA   ADD        SP0  CTF               NO; R31_R31+V1, SF1 IF POS
548           0611  30306150606                RD   JMP  ED32 SP1        NF1         ADD RK AGAIN IF R3 NEG,
549           0612  14627777771           RC   CTRL SUB        RC                    SP1_R32, Q3_Q3-CTR
550           0613  23177777761      ED34 SR   SM   ADD        BUS  ROS             READ WEXP-256 FROM SM+2
551                                     *
552                                     *    Q1,2,3=X,RB,RC   16X  16X  15X X
553                                     *    R3=SP0,SP1                    X 15X  16X
554                                     *    V=RA                          X 15X
555                                     *
556                                     *    CALC: Q4=R31.32/V1 CARRIED OUT 17 PLACES (10 REQUIRED);
557                                     *          IF Q4>=2**16 THEN Q4_ -1.
558                                     *
559                                     *    Q4=SP1   (X)  16X
560                                     *
561                                     *    ON COMPLETION DONE WITH V1, SP0,SP2,SP3,RA,RD EMPTY, F2 MODIFIED.
562                                     *
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (18)  ***  UBUS OR RBUS OR SL1 MISSING FROM DVSB
563           0614  33764112775      ED40 SP0  RA   DVSB SL1             NCRY      CALC Q4(0),
564           0615  37766360647                JMP  ED47                  UNC         JMP IF Q4>=2**16
565           0616  37772607765           RBUS      REPN                  20        SP0(1:15),SP1(0) (NCRY,NF2)
566           0617  33764332276           UBUS RA   DVSB SL1       INCT CTRM      SP1_Q4=R31,32/V1
567                                     *
568                                     *    F3= -WSGN, OPND=WEXP-256, X,RB,RC,SP1=0, SR=2, OVFL CLR.
569                                     *
570                                     *    F1_WSGN, RD_WEXP-2, RB,SP3,SP1,RA_W1F-4*4,
571                                     *    RESTORE ORG X, OPND_W ABS ADDR, F3,CTR_0, SR_3;
572                                     *    EXIT TO NOR3 TO FINISH NORMALIZATION, RND AND PACK.
573                                     *
574           0620  32537777777      ED50      RB   ADD        SP3                   SP3_Q2
575           0621  31317777217                RC   ADD        SP1  INSR             SP1_Q3, SR_3
576           0622  01677747457                SP1  ADD        RA   CF1  F3          RA_Q4,
577           0623  37777777477                     ADD             SF1               F1_WSGN
578           0624  23177777761           SR   SM   ADD        BUS  ROS             READ ORG X FROM SM+3
579           0625  26611637600                OPND ROM        RD   037600          RD_WEXP-2
580           0626  37772717166           X         REPN            CF3  07        F3_0,
581           0627  33640733276           UBUS RA   OASR SP1  RB   INCT CTRM        CTR_0,
582           0630  17677777777                SBUS ADD        RA                    RB(7:15),SP3,1,RA_W1F-4*4
583           0631  32641600777                RB   POMN       RB   000777          RB(0:6)_0 IN CASE Q1(0)=1
584           0632  23176777777                SM   INC        BUS  ROS             READ W ABS ADDR FROM SM+1
585           0633  26546360154                OPND JMP  NOR3 X              UNC     RESTORE X; NORM,RND,PACK
586                                     *
587                                     *    Q2>=2**16: R2_R2-2**16*V2,3; IF R2>=0 Q2,3,4_ -1;
588                                     *      IF Q2>2**16 THEN R2_R2+V1,2,3-V2,3, Q2_Q2-1,
```

```
                ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

589                                    *      IF R2>=0 Q2,3,4_ -1; IF R2<0 RETURN TO ED22.
590                                    *    RB(0),SP1=Q2, F2,RB(0,1:15)=R21(0),"1",R21(1:15),
591                                    *      RD,SP0=R13,14, RA,SP2,OPND=V1,2,3.
592                                    *    ED22 EXIT: RB_0=Q2 MOD 2**16, SP3,RC,SP0_R2, F1_0 (R2 NEG).
593                                    *
594           0634  26627567150      ED26 RD   OPND SUB        RC   CTF  F2          RC_R13-V3 (F1 IF "POS"),
595           0635  32771700000           RB   ROM                100000            =R22; UBUS_R21
596           0636  35527157156           UBUS SP2  CAD        SP3  CTF  NF1         SP3_R21-V2-(1 IF NF1),
597           0637  35527777145           RBUS SP2  SUB        SP3  CTF              =R21 (F1 IF POS)
598           0640  37766140644                     JMP  ED27                F1        Q2,3,4_ -1 IF R2 POS
599           0641  01646000561                SP1  JMP  ED22 RB           ZERO      RB_Q2, ADD BK IF 2**16
600           0642  25537507773           RA   SP3  ADD        SP3          CRRY      R2_R2+V1,2,3-V2,3, POS?
601           0643  37646360561                     JMP  ED22 RB           UNC       NO; Q2_Q2-1, ADD BK
602                                    *
603           0644  37647377777      ED27                CAD        RB                FLSE Q2_ -1
604           0645  23177777761      ED37 SR   SM   ADD        BUS  ROS              READ WEXP-256 FROM SM+2
605           0646  37627377777                     CAD        RC                    Q3_ -1
606           0647  37307377777      ED47                CAD        SP1                   Q4_ -1
607           0650  37766360620                     JMP  ED50                UNC
608                                    *
609                                    *    Q3>=2**16: R3_R3-2**16*V2; IF R3>=0 Q3,4_ -1;
610                                    *      IF Q3>2**16 THEN R3_R3+V1,2-V2, Q3_Q3-1, IF R3>=0 Q3,4_ -1;
611                                    *      IF R3<0 RETURN TO ED32.
612                                    *    RC(0),SP1=Q3, F2,RC(0,1:15)=R31(0),"1",R31(1:15),
613                                    *      SP0=R23, RA,SP2=V1,2.
614                                    *    ED32 EXIT: RC_0=Q3 MOD 2**16, SP0,RD_R3, F1_0 (R3 NEG).
615                                    *
616           0651  37617567435      ED36 SP0         ADD        RD   CF2  F2          RD_R23=R32, CF2,
617           0652  31771700000                RC   ROM                100000            UBUS_R31
618           0653  35327777156           UBUS SP2  SUB        SP0  CTF              SP0_R31-V2=R31 (F1 IF POS)
619           0654  37766140645                     JMP  ED37                F1        Q3,4_ -1 IF R3 POS
620           0655  01626000606                SP1  JMP  ED32 RC           ZERO      RC_Q3, ADD BK IF 2**16
621           0656  33337507775           SP0  RA   ADD        SP0          CRRY      R3_R3+V1,2-V2, POS?
622           0657  37626360606                     JMP  ED32 RC           UNC       NO; Q3_Q3-1, ADD BK
623           0660  37766360645                     JMP  ED37                UNC       YES; Q3,4_ -1
624                                    *
625                                    *    DIV BY ZERO: W_U, SET CCA, EXIT TO EFV1 TO CHECK TRAPS.
626                                    *    SP1=U ABS ADDR, RD=U2, RC=W ABS ADDR, CTR=2, F1=0, SR=2.
627                                    *    SP0_W ABS ADDR, SR_0, UBUS_STA.
628                                    *
629           0661  01176777561      EDZR SR   SP1  INC        BUS  ROD              READ U4
630           0662  01177777561           SR   SP1  ADD        BUS  ROD              READ U3
631           0663  26642360666                OPND JSB  EDZ2 RB           UNC       WRITE U4,3 AT RC+SR+1,RC+SR
632           0664  26733377052           RB   OPND IOR        SP2  CLSP           SP2_U4 IOR U3, SR_0
633           0665  01177777577                SP1  ADD        BUS  ROD              READ U1
634                                    *
635           0666  31136777541      EDZ2 SR   RC   INC        BSP0 WRD              SR=2: W4,3_RB,OPND=U4,3
636           0667  10177777437                ODWN ADD        BUS  DATA               =0: W2,1_RD,OPND=U2,1,
637           0670  37127377555           SP0       CAD        BSP0 WPD                   SP0_W ABS ADDR
638           0671  26177707437                OPND ADD        BUS  DATA RSB         RETURN IF SUBROUTINE
639                                    *
640           0672  26777417757                OPND ADD             CCA  NZRO        CCA ON U1, IF U1=0
641           0673  35773377650           RD   SP2  IOR             CCZ                CCZ ON U2 IOR U3,4
642           0674  24766360215                STA  JMP  EFV1                UNC       EFV1 WITH UBUS_STA
643                                    *
```

```
644                                *
645                                *         ENEG/ECMP
646                                *    ENTER VIA JMP TABLE IN SEC 7 WITH SR=4,
647                                *    RB=U DB REL ADDR (ECMP), RA=V DB REL ADDR.
648                                *    ENEG: EXIT TO ENEG WITH CONDITIONS LISTED AT ENEG.
649                                *    ECMP: S_S-2, CK U,V BOUNDS, COMPARE U WITH V.
650                                *    PACK MAY EXIT THROUGH ECP5, SETTING CCA ON SP2.
651                                *
652                                &0701 (10701)
653          0701  22117777573     ECMP RA   DB   ADD        BSP1 ROD             READ V1, SP1_V ABS ADDR
654          0702  23337777777          SM   ADD        SP0                       SP0_SM
655          0703  34766777274          SP1  DL   RNDT             INCT          V>=DL? CTR_2
656          0704  01766777275          SP0  SP1  RNDT             INCT          SM>=V? (= IN CASE ENEG)
657          0705  00766020747               CIR  JMP  ENEG             EVEN     JMP IF ENEG
658          0706  22117777572          RB   DB   ADD        BSP1 ROD             READ U1, SP1_U ABS ADDR
659          0707  26657777577               OPND ADD        RB   POP             RB_V1
660          0710  34766777574          SP1  DL   RNDT             POP           S_S-2; RD=V1, RB,RA=TOS
661          0711  01766777775          SP0  SP1  RNDT                           U>=DL? SM>=U?
662          0712  22136777571          PC   DB   INC        BSP0 ROD             READ V2, SP0_V2 ABS ADDR
663          0713  26637777317               OPND ADD        PC   HBF             RC_U1, F1_USGN
664          0714  30763127776          UBUS RD   XOR                  POS       U AND V SAME SIGN?
665          0715  00773357751          PC   CIR  IOR             CCA  NEXT      NO; CCA ON U, CIR SO NO CCE
666                                *
667          0716  01116777577               SP1  INC        BSP1 ROD             READ U2
668          0717  26737557777               OPND ADD        SP2       NF1        SP2_V2; U,V POS?
669          0720  31767417750          RD   RC   SUB             CCA  NZRO      NO, CMP V WITH U
670          0721  30767407751          RC   RD   SUB             CCA  ZERO      YES, CMP U WITH V
671          0722  37777757777                    ADD                  NEXT      DONE IF NZRO
672          0723  23176777757               SM   INC        BUS  WRS             MAKE SURE ORG (S-3)
673          0724  32177777437               RB   ADD        BUS  DATA              IN MEM IS TRUE COPY
674          0725  37136777575          SP0       INC        BSP0 ROD             READ V3
675          0726  26622360741               OPND JSB  ECP7 PC             UNC     RC_U2, CMP U2,V2 (1C JMP)
676          0727  01116777577               SP1  INC        BSP1 ROD             READ U3
677          0730  26737777777               OPND ADD        SP2                  SP2_V3
678          0731  23177777741          SR   SM   ADD        BUS  WRS             MAKE SURE ORG (S-2)
679          0732  33177777437               RA   ADD        BUS  DATA              IN MEM IS TRUE COPY
680          0733  37176777575          SP0       INC        BUS  ROD             READ V4
681          0734  26622360741               OPND JSB  ECP7 PC             UNC     RC_U3, CMP U3,V3 (1C JMP)
682          0735  01176777577               SP1  INC        BUS  ROD             READ U4
683          0736  26737777777               OPND ADD        SP2                  SP2_V4
684          0737  26622360741               OPND JSB  ECP7 RC             UNC     RC_U4, CMP U4,V4 (1C JMP)
685          0740  35777757757     ECP5      SP2  ADD                  CCA  NEXT      CCE IF U=V
686                                *
687          0741  35763017776     ECP7 UBUS SP2  XOR                  NZRO      U(N)=V(N)?
688          0742  35723707777               SP2  AND        SP2       RSB       YES, RETURN WITH SP2_0
689          0743  31767547156          UBUS RC   SUB             CTF  F1        CMP V WITH U IF U,V NEG
690          0744  35767777151          RC   SP2  SUB             CTF            ELSE CMP U WITH V
691          0745  00726140740               CIR  JMP  ECP5 SP2        F1        CCG IF OP1>OP2 (2C JMP)
692          0746  37777757677                    ADD                  CCL  NEXT      ELSE CCL
693                                *
694                                *
695                                *         ENEG
696                                *    ENTER FROM ECMP WITH V BOUNDS CHECKED,
697                                *    SP1=V ABS ADDR, OPND=V1, CTR=2, SR=4 WITH 4 TOS VALID.
698                                *
```

```
      ADDRESS   CONTENTS      LABL PBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

699           0747  26726000753     ENEG       OPND JMP  ENG4 SP2        ZERO       SP2_V1, JMP IF ZERO
700           0750  01177777557     ENG2       SP1  ADD         BUS  WRD
701           0751  35171700000                SP2  ROM         BUS  100000         S_S-1, COMPL SGN,
702           0752  00773357556           UBUS CIR  TOP                  POPA NEXT       CCA (CIR SO NO CCE)
703                                 *
704           0753  23136777757     ENG4       SM   INC         BSP0 WRS             MAKE SURE (S-3)
705           0754  30177777437                RD   ADD         BUS  DATA              IN MEM IS TRUE COPY
706           0755  01176777577                SP1  INC         BUS  ROD             READ V2
707           0756  37136777755           SP0       INC         BSP0 WRS             MAKE SURE (S-2)
708           0757  31177777437                RC   ADD         BUS  DATA              IN MEM IS TRUE COPY
709           0760  14177777574           SP1  CTRL ADD         BUS  ROD             READ V3
710           0761  26766010750                OPND JMP  ENG2              NZRO       JMP IF V2 NZRO
711           0762  37176777755           SP0       INC         BUS  WRS             MAKE SURE (S-1)
712           0763  32177777437                RB   ADD         BUS  DATA              IN MEM IS TRUE COPY
713           0764  14176777574           SP1  CTRL INC         BUS  ROD             READ V4
714           0765  26766010750                OPND JMP  ENG2              NZRO       JMP IF V3 NZRO
715           0766  26766010750                OPND JMP  ENG2              NZRO       JMP IF V4 NZRO
716           0767  37777757557                     ADD                  POPA NEXT       S_S-1, CCE IF V=ZERO
717                                 *
718                                 *
719                                 *     PSHM PUSHES ONE TOS REG INTO MEM, CHECKING INCR SM FOR STOV.
720                                 *     IF OVFL RESTORE V REL ADDR, EXIT TO BND2 (STOV); SP1=V ABS ADDR.
721                                 *
722           0770  23176777757     PSHM       SM   INC         BUS  WRS
723           0771  10177777437                ODWN ADD         BUS  DATA              PUSH TOS REG INTO MEM
724           0772  23767117762           Z    SM   CAD                   NCRY       Z>=SM+1?
725           0773  23476707237                SM   INC         SM   DCSP RSB       YES; INC SM, DCSP, RET
726           0774  37531601752                     ROM         SP3  BND2              DELETE IF 16/17 REPLACED
727           0775  22667777774           SP1  DR   SUB         PA                     RESTORE V REL ADDR
728           0776  37571601752                     ROM         RAP  BND2              EXIT TO BND2 (STOV)
729                                 *
730                                 % BND2  1752
731                                 % TRP0  3134
732                                 *
733                                 *
734                                 *     PSHM PATCH; REQUIRES REBURN OF 16/17 0-7.
735                                 *     RESTORE V REL ADDR IF STOV; -1L IF SEC 16/17 REPLACED.
736                                 *
737                                 *
738                                 *     RESTORE X PATCH; REQUIRES REBURN OF 16/17 0-7.
739                                 *     SR=3, OVFL CLR: DMUL/DDIV ZAN1,2 JMPS CHANGED TO ZAN3;
740                                 *       X_(SM+SR), RD_0, EXIT TO ZAN2.
741                                 *     IF SEC 16/17 REPLACED DMUL/DDIV JMP TO NEW ZAN1 PRECEDING ZAN2
742                                 *       WITH RD=0 (EG. SECOND DMUL JMP JMPS TO DDIV JMP), F3=DC,
743                                 *       WHERE ZAN1: READ X UNC, ZAN2: ... UNC, OPND JMP ZAN2 X UNC; -1L.
744                                 *
745                                 &0020
746           0020  23177777761     ZAN3 SR    SM   ADD         BUS  ROS             READ X FROM SM+3
747           0021  37617777777                     ADD         RD                   RD_0
748           0022  26546360231                OPND JMP  ZAN2 X            UNC        RESTORE X, JMP
```

```
  ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

 749
 750                           * 3/9/76 EIS-DEC
 751                           &1024
 752                           *SHIFT LEFT 1 DIGIT
 753                           *THIS SUBROUTINE REQUIRES THAT SP1 AND SP3 BE FREE
 754                           *F2 = "ONLY USE FOUR WORDS"
 755                           *IT ALSO USES F1
 756          1024  33761770000     L1D'       RA   ROMN            170000          WILL ANY DIGITS BE LOST?
 757          1025  16777407777                UBUS ADD                  ZERO
 758          1026  37777777517                     ADD             SCRY              IF SO, THEN SET CARRY
 759          1027  33777772773     L1D  RA    RA   ADD  SL1
 760          1030  16777772316          UBUS UBUS ADD  SL1        HBF             F1 _ %004000 AND RA
 761          1031  31317777777                RC   ADD       SP1
 762          1032  32537777777                RB   ADD       SP3
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING ( 9)  ***  UBUS OR SBUS OR SL1 MISSING FROM OASL (TASL ON /20)
 763          1033  33760372770           RD   RA   OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 764          1034  16760372777                UBUS OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 765          1035  16760372777                UBUS OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 766          1036  16660372337                UBUS OASL SL1  RA   FHB             HIGH BIT _ F1(BIT 4 OF ORIG
 767          1037  37617777777                     ADD       RD                   RBUS HAS LSW
 768          1040  25657577777                SP3  ADD       RB        NF2
 769          1041  01637707777                SP1  ADD       RC        RSB        RETURN IF F2(SHORT)
 770          1042  01637777777                SP1  ADD       RC
 771          1043  37775372760           PL        CRS  SL1
 772          1044  16775372777                UBUS CRS  SL1
 773          1045  16775372777                UBUS CRS  SL1
 774          1046  16775372317                UBUS CRS  SL1       HBF             F1 _ BIT 4 OF PL
 775          1047  16761600017                UBUS ROMN           000017          UBUS _ PL.(0:3)&CSL(4)
 776          1050  30517777776           UBUS RD   ADD       RD                   CORRECTED RD
 777          1051  37437777760           PL        ADD       Q
 778          1052  21237777777                Q    ADD       FL                   INTERCHANGE PL AND Q
 779          1053  22317777777                DB   ADD       SP1
 780          1054  34537777777                DL   ADD       SP3
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING ( 9)  ***  UBUS OR SBUS OR SL1 MISSING FROM OASL (TASL ON /20)
 781          1055  21760372760           PL   Q    OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 782          1056  16760372777                UBUS OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 783          1057  16760372777                UBUS OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 784          1060  16220372337                UBUS OASL SL1  PL   FHB
 785          1061  37437777777                     ADD       Q                    RBUS IS LSW
 786          1062  25717777777                SP3  ADD       DL
 787          1063  01457707777                SP1  ADD       DB        RSR
```

```
  788
  789                                *SHIFT RIGHT 1 DIGIT
  790                                *THIS SUBROUTINE USES SP1,SP3
  791           1064  31317777777     R1D       RC    ADD       SP1
  792           1065  32537777777               RB    ADD       SP3
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
  793           1066  30760773773          RA    RD    OASR SP1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
  794           1067  37760773776          UBUS        OASR SP1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
  795           1070  37760773776          UBUS        OASR SP1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
  796           1071  37660773776          UBUS        OASR SP1  RA
  797           1072  37617777777                      ADD       RD                  SBUS HAS LSW
  798           1073  30537774777               RD     ADD  RPZ  SP3                 SP3 _ ORIG RD AND %377
  799           1074  25657777777               SP3    ADD       RB
  800           1075  33661607777               RA     ROMN      RA    007777        MASK 4 MSB
  801           1076  01637577777               SP1    ADD       RC         NF2
  802           1077  37777707777                      ADD                 RSB       RETURN IF F2(SHORT)
  803           1100  22317777777               DR     ADD       SP1
  804           1101  34537777777               DL     ADD       SP3
  805           1102  25777772777               SP3    ADD  SL1                      SL1
  806           1103  16777772776          UBUS UBUS ADD  SL1                      SL2
  807           1104  16437775776          UBUS UBUS ADD  RLZ  0                    SL1,SL8 :ORIG RD&SL12
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
  808           1105  21760773760          PL    0     OASR SP1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
  809           1106  37760773776          UBUS        OASR SP1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
  810           1107  37760773776          UBUS        OASR SP1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
  811           1110  37220773776          UBUS        OASR SP1  PL
  812           1111  37437777777                      ADD       0                   SBUS HAS LSW
  813           1112  21537777777               0      ADD       SP3                 SP3 _ 0 (4 MSB OF FINAL PL)
  814           1113  25717777777               SP3    ADD       DL
  815           1114  01457777777               SP1    ADD       DR
  816           1115  37777777760          PL          ADD
  817           1116  16761607777               UBUS   ROMN           007777
  818           1117  25237707776          UBUS SP3    ADD       PL         RSR
```

```
  ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

                         *THIS SUBROUTINE SHIFTS LEFT BY 3 DIGITS
                         *IT USES SP1 AND SP3
                         *REG  0000 0000 0111 1111 1112 2222 2222 2333
                         *BEF  1234 5678 9012 3456 7890 1234 5678 9012
                         *
                         *SR1  X000 0000 0011 1111 1111 2222 2222 2233
                         *SR1  X123 4567 8901 2345 6789 0123 4567 8901
                         *
                         *SL3  0000 0011 1111 1111 2222 2222 2233 3XXX
                         *SL3  4567 8901 2345 6789 0123 4567 8901 2XXX
                         *
1120  33761407760  L3D'       RA   POMN          7760 ZERO      WILL ANY DIGITS BE LOST?
1121  37777777517                ADD             SCRY             IF SO, THEN SET CARRY
1122  32537777777  L3D         RB   ADD       SP3
1123  31317777777              RC   ADD       SP1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
1124  30760773773           RA   RD   OASR SP1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
1125  37760773776           UBUS      OASR SP1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
1126  37760773776           UBUS      OASR SP1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
1127  37760773776           UBUS      OASR SR1
1130  37637777777                ADD       RC
1131  01657777777           SP1  ADD       RB
1132  25677777777           SP3  ADD       RA
1133  30777772770           RD   RD   ADD  SL1                              SHIFT BY 2
1134  16777575777                UBUS ADD  RTZ            NF2              SHIFT BY 8
1135  16617702776           UBUS UBUS ADD  SL1  RD        RSB              SHIFT BY 2 IF RETURN,ELSE 1
1136  16617772777                UBUS ADD  SL1  RD                          SHIFT BY 1
1137  34537777777                DL   ADD       SP3
1140  22317777777                DB   ADD       SP1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
1141  21760773760           PL   0    OASR SR1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
1142  37760773776           UBUS      OASR SP1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
1143  37760773776           UBUS      OASR SR1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
1144  37700773776           UBUS      OASR SR1  DL                       DL _ MSW TEMPORARILY
1145  37457777777                     ADD       DB                       DB _ LSW
1146  25237777777                SP3  ADD       DL
1147  01717777777                SP1  ADD       DL
1150  34761607777                DL   POMN            007777             MSW AND %007777
1151  30617777776           UBUS RD   ADD       RD                       FIX RD
1152  21777775777                0    ADD  RTZ                            SHIFT BY 8
1153  16777772776           UBUS UBUS ADD  SL1                            SHIFT BY 2
1154  16437702776           UBUS UBUS ADD  SL1  0         RSB              SHIFT BY 2
```

```
 860
 861                                *THIS SUBROUTINE SHIFTS RIGHT BY 3 DIGITS
 862                                *IT USES SP1,SP3,F1
 863                                *REG  0000 0000 0111 1111 1112 2222 2222 2333
 864                                *REF  1234 5678 9012 3456 7890 1234 5678 9012
 865                                *
 866                                *SL1  0000 0000 1111 1111 1122 2222 2222 333X
 867                                *SL1  2345 6789 0123 4567 8901 2345 6789 012X
 868                                *
 869                                *SR3  XXX0 0000 0000 1111 1111 1122 2222 2222
 870                                *SR3  XXX1 2345 6789 0123 4567 8901 2345 6789
 871                                *
 872          1155  32537777777     R3D       RB   ADD        SP3
 873          1156  31317777777               RC   ADD        SP1
 874          1157  33775372777               RA   CRS  SL1                        CIRCULAR SHIFT LEFT 4
 875          1160  16775372777               UBUS CRS  SL1
 876          1161  16775372777               UBUS CRS  SL1
 877          1162  16775372317               UBUS CRS  SL1       HBF              F1 GETS RA.(4:1)
 878          1163  16661600017               UBUS ROMN       RA  000017           RA _ RA.(0:4)
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING ( 9)  ***  UBUS ON SBUS OR SL1 MISSING FROM OASL (TASL ON /20)
 879          1164  33760372770          RD   RA   OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 880          1165  16760372777               UBUS OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 881          1166  16760372777               UBUS OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 882          1167  16640372337               UBUS OASL SL1  PR   FHB
 883          1170  37317777777                    ADD       SP1                  SAVE LSW IN SP1
 884          1171  01617577777               SP1  ADD       RD        NF2
 885          1172  25637707777               SP3  ADD       PC        RSB        RETURN IF F2(SHOPT)
 886          1173  25637777777               SP3  ADD       PC
 887          1174  34537777777               DL   ADD       SP3
 888          1175  33775372760          PL        CRS  SL1                        CIRCULAR SHIFT LEFT 4
 889          1176  16775372777               UBUS CRS  SL1
 890          1177  16775372777               UBUS CRS  SL1
 891          1200  16775372317               UBUS CRS  SL1       HBF              F1 GETS PL.(4:1)
 892          1201  16701600017               UBUS ROMN       DL  000017           DL _ PL.(0:4)
 893          1202  22317777777               DB   ADD       SP1
 894          1203  34237777774          SP1  DL   ADD       PL                   FINAL PL _ PL&SR12 + RD&SL4
 895          1204  37437777760          PL        ADD       0                    PUT PL IN AN SBUS-REG
 896          1205  21777777777               0    ADD                            PUT 0 ON UBUS
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING ( 9)  ***  UBUS ON SBUS OR SL1 MISSING FROM OASL (TASL ON /20)
 897          1206  21760372776          UBUS 0    OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 898          1207  16760372777               UBUS OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 899          1210  16760372777               UBUS OASL SL1
***  WARNING ( 3)  ***  RBUS MAY BE FORCED ON FOLLOWING LINE
 900          1211  16700372337               UBUS OASL SL1  DL   FHB
 901          1212  37777777777                    ADD                            THROW AWAY LSW
 902          1213  01437777777               SP1  ADD       0
 903          1214  25457707777               SP3  ADD       DB        RSB
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN 4, 1976, 10:13 AM

904
905                                     *THIS SUBROUTINE SHIFTS LEFT 2 DIGITS
906                                     *ORIG 00 00 00 00 01 11 11 11
907                                     *ORIG 12 34 56 78 90 12 34 56
908                                     *
909                                     *SL2  00 00 00 00 11 11 11 1X
910                                     *SL2  23 45 67 89 01 23 45 6X
911             1215  33761407400      L2D'       RA    ROMN             7400 ZERO      WILL ANY DIGITS BE LOST?
912             1216  37777777517                 ADD                    SCRY             IF SO, THEN SET CARRY
913             1217  33777774777      L2D        RA    ADD   RPZ                       (0,2)
914             1220  32773371776            UBUS RB    IOR   LLZ                       (0,2) + (3,4) => (3,0)
915             1221  16677776765            RBUS UBUS  ADD   SWAP RA                   (3,0) + (0,2) => (2,3)
916             1222  32777774777                 RB    ADD   RPZ                       (0,4)
917             1223  31773371776            UBUS RC    IOR   LLZ                       (0,4) + (5,6) => (5,0)
918             1224  16657776765            RBUS UBUS  ADD   SWAP RB                  (0,4) + (5,0) => (4,5)
919             1225  31777774777                 RC    ADD   RPZ                       (0,6)
920             1226  30773371776            UBUS RD    IOR   LLZ                       (0,6) + (7,8) => (7,0)
921             1227  16637576765            RBUS UBUS  ADD   SWAP PC         NF2      (0,6) + (7,0) => (6,7)
922             1230  30617705777                 RD    ADD   RLZ  RD        RSB      (8,0)
923             1231  30777774777                 RD    ADD   RPZ                       (0,8)
924             1232  16773371760            PL   UBUS  IOR   LLZ                       (9,10) + (0,8) => (9,0)
925             1233  17617776776            UBUS SBUS  ADD   SWAP RD                  (9,0) + (0,8) => (8,9)
926             1234  37777774760            PL         ADD   RPZ                       (0,10)
927             1235  34773371776            UBUS DL    IOR   LLZ                       (0,10) + (11,12) => (11,0)
928             1236  16237776765            RBUS UBUS  ADD   SWAP PL                  (0,10) + (11,0) => (10,11)
929             1237  34777774777                 DL    ADD   RPZ                       (0,12)
930             1240  22773371776            UBUS DB    IOR   LLZ                       (0,12) + (13,14) => (13,0)
931             1241  16717776765            RBUS UBUS  ADD   SWAP DL                  (0,12) + (13,0) => (12,13)
932             1242  22777774777                 DB    ADD   RPZ                       (0,14)
933             1243  21773371776            UBUS 0     IOR   LLZ                       (0,14) + (15,16) => (15,0)
934             1244  16457776765            RBUS UBUS  ADD   SWAP DB                  (0,14) + (15,0) => (14,15)
935             1245  21437705777                 0     ADD   RLZ  0         RSB
```

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

936
937                             *THIS SUBROUTINE SHIFTS RIGHT 2 DIGITS
938                             *ORIG 00 00 00 00 01 11 11 11
939                             *ORIG 12 34 56 78 90 12 34 56
940                             *
941                             *SR2  X0 00 00 00 00 11 11 11
942                             *SR2  X1 23 45 67 89 01 23 45
943          1246  33677770777  R2D        RA    ADD  LRZ  RA                          (0,1)
944          1247  33777774777             RA    ADD  RRZ                             (0,2)
945          1250  32773371776        UBUS RB    IOR  LLZ                             (0,2) + (3,4) => (3,0)
946          1251  16657776765        RBUS UBUS  ADD  SWAP RB                          (3,0) + (0,2) => (2,3)
947          1252  32777774777             RB    ADD  RRZ                             (0,4)
948          1253  31773371776        UBUS RC    IOR  LLZ                             (0,4) + (5,6) => (5,0)
949          1254  16637776765        RBUS UBUS  ADD  SWAP RC                          (0,4) + (5,0) => (4,5)
950          1255  31777774777             RC    ADD  RRZ                             (0,6)
951          1256  30773171776        UBUS RD    IOR  LLZ              NF2            (0,6) + (7,8) => (7,0)
952          1257  16617706765        RBUS UBUS  ADD  SWAP RD          RSB            (0,6) + (7,0) => (6,7)
953          1260  16617776777             UBUS  ADD  SWAP RD
954          1261  30777774777             RD    ADD  RRZ                             (0,8)
955          1262  16773371760        PL   UBUS  IOR  LLZ                             (9,10) + (0,8) => (9,0)
956          1263  17237776776        UBUS SBUS  ADD  SWAP PL                          (9,0) + (0,8) => (8,9)
957          1264  37777774760        PL         ADD  RRZ                             (0,10)
958          1265  34773371776        UBUS DL    IOR  LLZ                             (0,10) + (11,12) => (11,0)
959          1266  16717776765        RBUS UBUS  ADD  SWAP DL                          (0,10) + (11,0) => (10,11)
960          1267  34777774777             DL    ADD  RRZ                             (0,12)
961          1270  22773371776        UBUS DR    IOR  LLZ                             (0,12) + (13,14) => (13,0)
962          1271  16457776765        RBUS UBUS  ADD  SWAP DR                          (0,12) + (13,0) => (12,13)
963          1272  22777774777             DR    ADD  RRZ                             (0,14)
964          1273  21773371776        UBUS 0     IOR  LLZ                             (0,14) + (15,16) => (15,0)
965          1274  16437706765        RBUS UBUS  ADD  SWAP C           PSB            (0,14) + (15,0) => (14,15)
```

	ADDRESS	CONTENTS	LABL	PBUS	SBUS	FUNC	SHFT	STOR	SPEC	SKIP	COMMENTS
966											
967	1275	32337777537	DMPY		RB	ADD		SP0	CCRY		
968	1276	33317777777			RA	ADD		SP1			
969	1277	33537777777			RA	ADD		SP3			
970	1300	37772607777				REPN				20	
971	1301	16774333271		RC	UBUS	MPAD	SR1		INCT	CTRM	
972	1302	17657777777			SBUS	ADD		RB			
973	1303	25677777777			SP3	ADD		RA			
974	1304	37537777775		SP0		ADD		SP3			
975	1305	32772607777			RB	REPN				20	
976	1306	16774333271		RC	UBUS	MPAD	SR1		INCT	CTRM	
977	1307	17637777777			SBUS	ADD		PC			
978	1310	01537777777			SP1	ADD		SP3			
979	1311	25772607777			SP3	REPN				20	ADD IN LSW FROM PREV MULT
980	1312	16774333270		RD	UBUS	MPAD	SR1		INCT	CTRM	
981	1313	17317777777			SBUS	ADD		SP1			
982	1314	25657777777			SP3	ADD		RB			
983	1315	37537777775		SP0		ADD		SP3			
984	1316	01772607777			SP1	REPN				20	
985	1317	16774333270		RD	UBUS	MPAD	SR1		INCT	CTRM	
986	1320	17617777757			SBUS	ADD		RD	CCA		
987	1321	25637517771		RC	SP3	ADD		PC		NCRY	
988	1322	30616777757			RD	INC		RD	CCA		CCE,CCG,CCL
989	1323	16777537777			UBUS	ADD				NEG	SKIP IF CCL
990	1324	30773007651		PC	RD	IOR			CCZ	ZERO	IF CCE,CCG
991	1325	37777757517				ADD			SCRY	NEXT	IF CCL OR CCG
992	1326	33773357652		RB	RA	IOR			CCZ	NEXT	CCG OR CCE

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

 993
 994                            * THIS SUBROUTINE DETERMINES IF IT IS NECESSARY TO FREE FOUR
 995                            * ADDITIONAL REGISTERS (Q DL DB PL)
 996                            * FRAD IS A SPECIAL ENTRY FOR ADDD/SUBD/CMPD
 997                            * FRLG IS AN ENTRY TO FREE THEM UNCONDITIONALLY
 998     1327  33771537762      FRAD       RA    ROM              7762 NEG        LONG IF BLEN>13
 999     1330  37766171335                 JMP   FRLG                NF2        AND A-HALF-WORD
1000     1331  25771527774      FREE       SP3   ROM              7774 POS        A-WDCNT > 3?
1001     1332  14771537774      FREB       CTRL  ROM              7774 NEG        B-WDCNT > 3?
1002     1333  37777767437                 ADD                    CF2  UNC
1003     1334  37777707417                 ADD                    SF2  RSB        BOTH OPERANDS <= 4WDS EACH
1004     1335  23771600005      FRLG       SM    ROM              000005
1005     1336  16137777741           SR    UBUS  ADD        PSP0  WRS             FINAL SM+5
1006     1337  37177777420           PL          ADD        BUS   DATA
1007     1340  37136777755           SP0         INC        PSP0  WRS             +6
1008     1341  34177777437                 DL    ADD        BUS   DATA
1009     1342  37136777755           SP0         INC        PSP0  WRS             +7
1010     1343  22177777437                 DB    ADD        BUS   DATA
1011     1344  37136777755           SP0         INC        PSP0  WRS             +8
1012     1345  21177707437                 0     ADD        BUS   DATA RSB
1013                            *IF THERE IS A SPLIT STACK
1014                            *IT RECALCULATES SP2 AND SP1
***  WARNING ( 2)  ***  RBR CONFLICTS WITH PREFETCH ON INSTR ENTRY
***  WARNING ( 1)  ***  RBR MAY CONFLICT WITH PREVIOUS BANK SELECTION
1015     1346  03777777417      SPLT       RBR   ADD              DB
1016     1347  03767417616           UBUS  RBR   SUB              S     NZRO      DB-BANK = S-BANK?
1017     1350  22767507762           Z     DB    SUB                    CRRY      Z > DB?
1018     1351  22766361355                 DB    JMP   SPT1               UNC       NO OR NO;SPLIT STACK
1019     1352  34767517776           UBUS  DL    SUB                    NCRY      DB > DL ?
1020     1353  37777707777                 ADD                          RSB       YES -- RETURN
1021     1354  35777777777                 SP2   ADD                                UBUS_SP2 FOR SLOW RSB
1022     1355  30777773777      SPT1       RD    ADD   SP1
1023     1356  22317777776           UBUS  DB    ADD         SP1                    RECALULATE SP1
1024     1357  32777773777                 RB    ADD   SP1
1025     1360  22737707776           UBUS  DB    ADD         SP2        RSB         RECALULATE SP2
1026                            *THIS SUBROUTINE CREATES THE NINES COMPLEMENT
1027     1361  37771714631      9CMP       ROM              114631          '9999'
1028     1362  33667777076           UBUS  RA    SUB         RA    SF3               SET F3
1029     1363  32647777765           RBUS  RB    SUB         RB
1030     1364  31627777765           RBUS  RC    SUB         RC
1031     1365  30607577765           RBUS  RD    SUB         RD          NF2
1032     1366  37777707760           PL          ADD                     PSB         EXIT IF F2(SHOPT)
1033     1367  16767777777                 UBUS  SUB
1034     1370  16231714631                 UBUS  ROM         PL    114631
1035     1371  21427777765           RBUS  0     SUB         0
1036     1372  34707777765           RBUS  DL    SUB         DL
1037     1373  22447707765           RBUS  DB    SUB         DB          RSB
1038                            *THIS SUBROUTINE PUSHES ALL FOUR STACK REGISTERS INTO CORE
1039                            *IT ALSO CLEARS OVERFLOW AND SR
1040     1374  23176777757      PSHA       SM    INC         BUS   WRS             PUSH REG
1041     1375  10177777437                 ODWN  ADD         BUS   DATA
1042     1376  23476657237                 SM    INC         SM    DCSP SRL2
1043     1377  37766361374                       JMP   PSHA               UNC
1044     1400  37777707637                       ADD                    CLO   RSB
```

```
1045
1046                                *THIS IS THE SUBROUTINE THAT SCANS 'A' AND 'B' FOR SIGNIFICANT DIGITS
1047                                *THE ENTRY POINTS ARE DIFFERENT FOR 'A' AND 'B' BUT THEY RETURN FROM
1048                                *A COMMON SECTION OF CODE.  THE NUMBER OF SIGNIFICANT DIGITS IS RETURNED
1049                                *IN SP0
1050                                *F1 SHOULD CONTAIN THE LSB OF RB UPON ENTRY
1051          1401  35177777577     SCNB      SP2  ADD        BUS  ROD              READ MSW
1052          1402  33675032777               RA   CRS  SL1  RA        ODD         RESTORE RA-MSKB DID CRS SR1
1053          1403  16676777777               UBUS INC       RA                    MAKE RA ODD IF IT'S NOT NOW
1054          1404  36337557777               RB   ADD       SP0       NF1         F1 IS LSB OF RB;SP0_MSWMASK
1055          1405  33671600002               RA   ROM       RA   000002          RA _ WHAT IT WOULD BE IF
1056                                *                                               IT HAD 4 DIGITS IN MSW
1057          1406  26163417635         SP0   OPND AND       BUS  OPND NZRO       MSW AND MSW-MASK
1058          1407  14777417777               CTRL ADD                 NZRO        LAST WORD ?
1059          1410  33326361421               RA   JMP  SCAN SP0       UNC         FOUND ONE;SP0 _ RA
1060          1411  37747377777                    CAD       RB                    MSW =0;MSW-MASK _ -1
1061          1412  35176777577     SCB1      SP2  INC       BUS  ROD
1062          1413  35736777777               SP2  INC       SP2                   UPDATE @MSW
1063          1414  33671777774               RA   ROM       RA   177774          UPDATE DIGIT CNT
1064          1415  14351417777               CTRL ROM       CTRL 7777 NZRO       UPDATE WORD CNT;0?
1065          1416  33326361421               RA   JMP  SCAN SP0       UNC         LAST WORD;CTRL IS 0
1066          1417  26766001412               OPND JMP  SCB1           ZERO        TRY AGAIN IF OPND=0
1067          1420  33337777777               RA   ADD       SP0                   SET SP0 TO DIGIT CNT
1068          1421  26761770000     SCAN      OPND ROMN                170000
1069          1422  16777407777               UBUS ADD                 ZERO        FIRST DIGIT NON-ZERO?
1070          1423  37777707777                    ADD                 RSB         YES
1071          1424  37327007775         SP0        CAD       SP0       ZERO        DECR DIGIT CNT;IS IT 0?
1072          1425  26761407400               OPND ROMN                7400 ZERO   SECOND DIGIT NON-ZERO?
1073          1426  37777707775         SP0        ADD                 RSB         YES
1074          1427  37327377775         SP0        CAD       SP0                   DECR DIGIT CNT
1075          1430  26761410360               OPND ROMN                0360 NZRO   THIRD DIGIT NON-ZERO?
1076          1431  37327307775         SP0        CAD       SP0       RSB         NO
1077          1432  37777707776         UBUS       ADD                 RSB         YES
1078          1433  01177777577     SCNA      SP1  ADD       BUS  ROD              READ MSW
1079          1434  31635032777               RC   CRS  SL1  RC        ODD         RESTORE RC-MSKA DID SR1
1080          1435  16636777777               UBUS INC       RC                    MAKE RC ODD IF IT ISN'T
1081          1436  30777427777               RD   ADD                 EVEN
1082          1437  31631600002               RC   ROM       RC   000002          MAKE RC WHAT IT WOULD BE IF
1083                                *                                               THE MSW WERE 4 DIGITS
1084          1440  26163417622         Z     OPND AND       BUS  OPND NZRO       MSW 0?
1085          1441  25521410007               SP3  ROMN       SP3  0007 NZRO       LAST WORD?(ROMN FOR MPYD)
1086          1442  31326361421               RC   JMP  SCAN SP0       UNC         FOUND IT;SP0 _ RC
1087          1443  37247377777                    CAD       7                     MAKE MSW-MASK -1
1088          1444  01116777577     SCA1      SP1  INC       BSP1 ROD              UPDATE @MSW
1089          1445  31631777774               RC   ROM       RC   177774          UPDATE DIGIT CNT
1090          1446  25531417777               SP3  ROM       SP3  7777 NZRO       UPDATE WORD CNT;0?
***  WARNING ( 8)  ***  TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
1091          1447  31326361421               RC   JMP  SCAN SP0       UNC         LAST WORD
1092          1450  26766001444               OPND JMP  SCA1           ZERO
1093          1451  31326361421               RC   JMP  SCAN SP0       UNC
```

```
1094
1095                          *THESE ARE THE TRAPS FOR THE ADDO FAMILY OF INSTRUCTIONS
1096          1452  37531520017 TA17            ROM         SP3  0017 POS
1097          1453  37531520013 TX13            ROM         SP3  0013 POS       SKIP PSHA
1098          1454  37762361374                 JSR  PSHA             UNC       THIS JSB OVERRIDES ANY PREV
1099          1455  37766361461                 JMP  TAUT             UNC
1100          1456  37531520015 TA15            ROM         SP3  0015 POS
1101          1457  37531600013 TA13            ROM         SP3  000013
1102          1460  37762361465                 JSR  RSTA             UNC       RESTORE REGISTERS
1103          1461  24777777057 TAUT      STA   ADD                   CLSR      INTERRUPT TRAPS ENABLED?
1104          1462  16777522616       UBUS UBUS ADD  SL1            SOV  POS
1105          1463  37571603134                 ROM         RAP  IRP0
1106          1464  37766362220                 JMP  APOP             UNC
```

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

1107
1108                                *THIS SUBROUTINE RESTORES THE 3 OR 7 REGISTERS FREED BY FREE
1109      1465  23776777777     RSTA       SM    INC
1110      1466  16136777777                UBUS  INC         BSP0  ROS               SM+2
1111      1467  16136777777                UBUS  INC         BSP0  ROS               SM+3
1112      1470  26557777777                OPND  ADD         X
1113      1471  37136777775          SP0         INC         BSP0  ROS               SM+4
1114      1472  26757777777                OPND  ADD         PB
1115      1473  37136577775          SP0         INC         BSP0  ROS   NF2         SM+5
1116      1474  26257707777                OPND  ADD         Z           RSB         RETURN IF SHORT
1117      1475  16246361500                UBUS  JMP   *+3   Z           UNC
1118      1476  23331600005     RSTL       SM    ROM         SP0   000005
1119      1477  16177777777                UBUS  ADD         BUS   ROS               SM+5
1120      1500  37136777775          SP0         INC         BSP0  ROS               SM+6
1121      1501  26237777777                OPND  ADD         PL
1122      1502  37136777775          SP0         INC         BSP0  ROS               SM+7
1123      1503  26717777777                OPND  ADD         DL
1124      1504  37136777775          SP0         INC         BSP0  ROS               SM+8
1125      1505  26457777777                OPND  ADD         DB
1126      1506  26437707777                OPND  ADD         0           RSB
```

```
1127
1128                                *THIS SUBROUTINE FETCHES THE 'B' OPERAND.
1129                                *UPON ENTRY IT EXPECTS:
1130                                *SP2 - WORD ADDRESS OF MSW
1131                                *UBUS _ WORD COUNT (0-7)
1132                                *F2 _ "FETCH ONLY FOUR WORDS"
1133                                *RB _ SIGN MASK
1134                                *PB _ MSW MASK
1135                                *X _ SIGN MASK OF 'A'
1136                                *IT USES SP0,F1,CTRL,SP2 ( AND THEY ARE FREE AT THE END )
1137                                *IT RETURNS:
1138                                *FOUR OR EIGHT WORDS IN (RA,RB,RC,RD) OR (RA,RB,RC,RD,PL,DL,DB,Q)
1139                                *F3 _ "DON'T NEED TO SHIFT 'B' TO LINE UP WITH 'A'"
1140                                *ABS _ INCREMENTED IF 'B' IS NEGATIVE
1141                                *THE SIGN OF B IS ALSO SET IN STA (CCL OR CCG)
1142     1507  35137777576  FTCH UBUS SP2  ADD        FSP0 ROD             SP0 _ @LSW
1143     1510  32763417066       X   RB   AND             SF3  NZRO       DO SIGN MASKS AGREE?
1144     1511  37777777177            ADD                 SF3             F3="DON'T NEED TO SHIFT 'B'
1145     1512  32767057677            RB   CAD            CCL  BIT6       RB _ 'FOFF' OR 'FFF0'
1146     1513  16777771777            UBUS ADD  LLZ                        UBUS WAS 'FOFF';MAKE 'F000'
1147     1514  26663777776       UBUS OPND AND        RA                   RA _ LSW
1148     1515  17643577452       RB   SBUS AND        RB   CF1  NF2        RB _ SIGN DIGIT
1149     1516  37777777477            ADD                 SF1             SF1 IF "SHORT"(IE,F2)
1150     1517  37127377575       SP0       CAD        FSP0 ROD             FETCH 2ND WORD
1151     1520  32761410017            RB   ROMR               0017 NZRO    IN POSITION '000X'? YES=OK
1152     1521  17777776777            SBUS ADD  SWAP                       NO=ROTATE(ALSO IF ALL 0)
1153     1522  16761010015            UBUS ROMX               0015 NZRO    B-SIGN NEG?
1154     1523  03156767017            RBR  INC        SBR  ABS  UNC        INCR ABS=BANK=REG IF NEG
1155     1524  37777777717            ADD                 CCG
1156     1525  37351777774  FTA'      RB4        CTRL 177774              CTRL _ -4
1157     1526  35731527777  FTAC      SP2  ROM        SP2  7777 POS       DECR WD CNT
1158     1527  37766361541            JMP  FTDN                 UNC       JUMP DONE IF WD CNT -
1159     1530  37127377575       SP0       CAD        FSP0 ROD
1160     1531  26777737277            OPND ADD                 INCT CTRM
1161     1532  16206361526            UBUS JMP  FTAC PUSH      UNC       JUMP AGAIN UNLESS CTRM
1162     1533  16677557777            UBUS ADD        RA       NF1       THIS IS 5TH WD => TOS
1163     1534  33677707057            RA   ADD        RA   CLSR RSB       SPECIAL RETUEN FOR MPYD
1164     1535  16237777777            UBUS ADD        PL                   4TH WD => PL
1165     1536  32717777477            RB   ADD        DL   SF1             3RD WD => DL;DONE PL-Q
1166     1537  31457777777            RC   ADD        DB                   2ND WD => DB
1167     1540  30426361525            RD   JMP  FTA'  Q            UNC       JUMP AGAIN
1168     1541  36663777773  FTDN PA   PB   AND        PA                   LAST WORD FETCHED AND MSWM
1169     1542  23771600003            SM   ROM                000003       S+3
1170     1543  16177777777            UBUS ADD        BUS  ROS             FETCH PB
1171     1544  37772337277                 REPC               INCT CTR4    SKIP NEXT LINE IF CTRM
1172     1545  37217737277                 ADD        PUSH INCT CTRM       ZERO REST OF RA-RD
1173     1546  37617557777                 ADD        RD        NF1        ZERO RD IF NF1
1174     1547  30617707057            RD   ADD        RD   CLSR RSB       RESTORE RD AND EXIT IF F1
1175     1550  16437777777            UBUS ADD        Q
1176     1551  31457777777            RC   ADD        DB
1177     1552  32717777777            RB   ADD        DL
1178     1553  33237777777            RA   ADD        PL
1179     1554  37637777777                 ADD        PC
1180     1555  37657777777                 ADD        PB
```

	ADDRESS	CONTENTS	LABL	RBUS	SBUS	FUNC	SHFT	STOR	SPEC	SKIP	COMMENTS
1181	1556	37677707057				ADD		RA	CLSR	RSB	
1182			*								
1183			*								
1184			*								
1185			* WORDS FETCHED			F2	NF2				
1186			*	1		28	35				
1187			*	2		32	39				
1188			*	3		36	43				
1189			*	4		40	47				
1190			*	5			55				
1191			*	6			59				
1192			*	7			63				
1193			*	8			67				

```
1194
1195                                *THIS SUBROUTINE STORES 1-8 REGISTERS WITH INFORMATION IN
1196                                *DECIMAL FORMAT AND USES THE NECESSARY MASKS TO PRESERVE
1197                                *UNUSED PARTS OF THE WORDS.
1198                                *IT ALSO SETS OVERFLOW AND SETS THE PROPER SIGN INTO THE FIELD.
1199                                *UPON ENTRY IT EXPECTS:
1200                                *SP1 - @MSW IN MEMORY
1201                                *SP3 - WORD COUNT (0-7)
1202                                *F2 - "USE ONLY FOUR WORDS"
1203                                *X - SIGN MASK
1204                                *Z - MSW MASK
1205                                *IT ALSO USES SP0,SP2
1206                                *IF NECESSARY, IT WILL SET CCE AND MAKE THE STORED SIGN POSITIVE IF
1207                                *ZERO (+ OR -) IS STORED IN THE AVAILABLE FIELD (IE, OVERFLOW CAN ALSO
1208                                *BE SET IF NON-STORED DIGITS ARE NOT ZERO).
1209                                *AT THE END, F1 = "NON-STORED DIGITS ARE NON-ZERO" (IF,OVERFLOW)
1210                                *F1 SHOULD BE CLEARED BEFORE ENTRY UNLESS OVERFLOW HAS ALREADY
1211                                *BEEN DETECTED.
1212                                *SP2 SHOULD BE CLEARED IN THE CALL (I.E., JSB STOR SP2 UNC)
1213      1557  37311777774   STOR            ROM       SP1  177774          SET SP1 _ -4 IF F2
1214      1560  01137567577           SP1     ADD       BSP0 ROD  F2         SP0 _ @MSW;FETCH MSW
1215      1561  37311777770           ROM       SP1  177770          SP1 _ -8 IF NF2
1216      1562  37351777774           ROM       CIRL 177774
1217      1563  25776417774   STSC SP1  SP3     INC                   NZRO
1218      1564  37766361574            JMP  STMS                      UNC        JUMP TO MSW PART
1219      1565  35733377763      MREG SP2  IOR       SP2               ACCUM. LEADING WDS IN SP2
1220      1566  01316737277           SP1     INC       SP1  INCT CTRM        BUMP MREG POINTER
1221      1567  37766361563           JMP  STSC                      UNC        NO JUMP TO SCAN AGAIN
1222      1570  37677777760         PL        ADD       RA                    YES,FILL ARRAY WITH NEW WDS
1223      1571  34657777777               DL  ADD       RB
1224      1572  22637777777               DR  ADD       RC
1225      1573  21606361563               0   JMP  STSC RD              UNC        JUMP SCAN AGAIN
1226      1574  37777777762   STMS Z            ADD                               Z IS MSW MASK
1227      1575  16762777763        MREG UBUS CAND                               UBUS IS MSW AND NOT(MSW=MAS
1228      1576  35733007776        UBUS SP2  IOR       SP2               ZERO       SP2 _ TOTAL LEADING DIGITS
1229      1577  37777777462        Z         ADD                 SF1              SOV IF LEADING DIGITS <> 0
1230      1600  16723777763        MREG UBUS AND       SP2                        SP2 _ MSW DIGITS TO BE STOR
1231      1601  17767377777             SBUS CAD                               UBUS _ NOT(MSW=MASK)
1232      1602  26763777776        UBUS OPND AND                                KEEP PART OF MSW IN MEM
1233      1603  35073377776        UBUS SP2  IOR       MREG                     MASKED AND MERGED MSW TO MR
1234      1604  25177777575        SP0  SP3  ADD       BUS  ROD                 PREFETCH SIGN WORD
1235      1605  37327367775        SP0       CAD       SP0               UNC        SKIP NEXT LINE 1ST TIME
1236      1606  35733377763   STRS MREG SP2  IOR       SP2                        ACCUM.STORED DIGITS IN SP2
1237      1607  01777447777           SP1     ADD                         NSME
1238      1610  37766361621           JMP  STSN                      UNC        JUMP SIGN IF SP1=%177777
1239      1611  37136777555        SP0       INC       BSP0 WRD
1240      1612  37177777423        MREG      ADD       BUS  DATA
1241      1613  01316737277           SP1     INC       SP1  INCT CTRM        BUMP POINTER
1242      1614  37766361606           JMP  STRS                      UNC        NO JUMP REST AGAIN
1243      1615  37677777760         PL        ADD       RA                    YES FILL ARRAY WITH NEW WDS
1244      1616  34657777777               DL  ADD       RB
1245      1617  22637777777               DR  ADD       RC
1246      1620  21606361606               0   JMP  STRS RD              UNC        RETURN TO REST AGAIN
```

```
1247
1248                                *THIS SECTION STORES THE SIGN WORD
1249          1621  23311600003     STSN       SM    FOM         SP1  000003
1250          1622  16176777777                UBUS  INC         BUS  ROS              GET Z
1251          1623  26537774777                OPND  ADD   RPZ   SP3                   IF '0F00'
1252          1624  35777417777                SP2   ADD                     NZRO       ANY STORED DIGITS <> 0?
1253          1625  37777777737                      ADD                   CCE
1254          1626  37731600014                      FOM         SP2  000014           '000C'
1255          1627  24761400400                STA   FOMN                  0400 ZERO    CCL ?
1256          1630  37731600015                      FOM         SP2  000015           YES,'000D'
1257          1631  37167377774           SP1        CAD         BUS  ROS              GET X
1258          1632  26257777777                OPND  ADD         7
1259          1633  37777427766           X          ADD                     EVEN       '0F00' ?
1260          1634  37537767777                      ADD         SP3         UNC        NO,CLEAR SP3 IF X=000F
1261          1635  35737776777                SP2   ADD   SWAP  SP2                   YES,ROTATE SIGN INTO PLACE
1262          1636  25613377770           RD   SP3   IOR         RD                    RD OR RD IOR LSW LLZ
1263          1637  37136777555           SP0        INC         BSP0 WPD
1264          1640  35173307430           RD   SP2   IOR         BUS  DATA RSB         RD IOR SIGN
1265                                      *
1266                                      *
1267                                      *
1268                                      * LEADING WORDS  F2  NF2
1269                                      *               0   58  90
1270                                      *               1   56  88
1271                                      *               2   54  86
1272                                      *               3   52  84
1273                                      *               4       82
1274                                      *               5       80
1275                                      *               6       78
1276                                      *               7       76
```

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:13 AM

1277
1278      1641  23311600007  CVAD       SM   ROM        SP1  000007        SET SP1 FOR CKA'
1279      1642  16767507762       %    UBUS SUB                    CRRY
1280      1643  37571601752            ROM        RAR  BND2          STOV
1281      1644  23322361374            SM   JSB  PSHA SP0          UNC   RA:SCNT,RB:SAD
1282      1645  33771507743            RA   ROM             7743 CRRY  RC:TCNT,RD:TAD
1283      1646  31777517625       RBUS RC   ADD             CLO  NCRY
1284      1647  37766363313            JMP  TF17                  UNC
1285      1650  33766003302            RA   JMP  BPOP             ZERO
1286      1651  31766003302            RC   JMP  BPOP             ZERO
1287      1652  37762362062            JSB  CKA'                  UNC
1288      1653  37767377773       RA        CAD
1289      1654  33677667776       UBUS RA   ADD        RA         NPRV  DOUBLE FOR CKBB
1290      1655  37762361346            JSB  SPLT                  UNC   SPLIT STACK?
1291      1656  25177577574       SP1  SP3  ADD        BUS  ROD   NF2   FETCH SIGN WORD OF 'A'
1292      1657  37777777077            ADD             SF3          F3 _ F2 (IF,A FULL WORD)
1293      1660  37762362041            JSB  CKB'                  UNC
1294      1661  24762131346            STA  JSB  SPLT             NEG   FIX SP2 IF SPLIT STACK
1295      1662  37351777774            ROM        CTRL 177774         CNTR _ -4
1296      1663  14646341667            CTRL JMP  CAD1 PB          F3    JMP IF SIGN'A IS RT BYTE
1297      1664  26777774777            OPND ADD  RLZ                  GARBAGE BYTE
1298      1665  16777772276       UBUS UBUS ADD  SL1        INCT
1299      1666  16317762276       UBUS UBUS ADD  SL1  SP1   INCT UNC   RIGHT BYTE &SL4 TO SP1
1300      1667  01307777774  CAD1 SP1  SP1  SUB        SP1              CLEAR SP1
1301      1670  25737777465       RBUS SP3  ADD        SP2  SF1         SP2 _ @A-SIGN WORD
1302      1671  35137777572       PB   SP2  ADD        BSP0 ROD         SP0_@B-SIGN-WORD
1303      1672  33667373337            RA   CAD  SP1  RA   FFB         NEG ASCII CNT TO RA
1304      1673  01311350000            SP1  POMI       SP1  150000      NEGATIVE SIGN
1305      1674  31627567677            RC   SUB        RC   CCL  F2     NEG DECIMAL CNT
1306      1675  37127367575       SP0       CAD        BSP0 ROD  UNC    NF2:LEFT BYTE TO SP3;GET 1
1307      1676  26777775777            OPND ADD  RLZ                    F2: RIGHT BYTE TO SP3
1308      1677  16537770457            UBUS ADD  LFZ  SP3  SF1          BOTH:CLEAR 'BLANKMODE'
1309      1700  16531417603            UBUS ROM        SP3  7603 NZRO   -%175;SP3 HAS @'FD BYTE
1310      1701  37606361744            JMP  CAD7 RD               UNC   NEGATIVE ZERO
1311      1702  37617777717            ADD       RD   CCG
1312      1703  01301747777            SP1  POMN       SP1  147777      POSITIVE SIGN
1313      1704  25531410002            SP3  ROM        SP3  0002 NZRO   %173
1314      1705  37766361744            JMP  CAD7                  UNC   POSITIVE ZERO(%173)
1315      1706  25531530050            SP3  ROM        SP3  0050 NEG     7655
1316      1707  37766363316            JMP  TF14                  UNC   >=123 (AND NOT %173&5)IS TP
1317      1710  25531530011            SP3  ROM        SP3  0011 NEG     7666
1318      1711  37766361725            JMP  CAD3                  UNC   %112 - %122 (SET NEG & INC)
1319      1712  25531530011            SP3  ROM        SP3  0011 NEG     7677
1320      1713  37766361727            JMP  CAD4                  UNC   %101 - %111 ( IS POS - INC)
1321      1714  25531530007            SP3  ROM        SP3  0007 NEG     7706
1322      1715  37766363316            JMP  TF14                  UNC   %072 - %100 ARE ILLEGAL
1323      1716  01311370000            SP1  POMI       SP1  170000      SET ABSOLUTE SIGN
1324      1717  25531530012            SP3  ROM        SP3  0012 NEG     7720
1325      1720  37766361743            JMP  CAD6                  UNC   %060 - %071 ARE 0-9(OK NOW)
1326      1721  25531400020            SP3  ROM        SP3  0020 ZERO    %40?(IF YES, THEN ABS 0)
1327      1722  37766363316            JMP  TF14                  UNC   NO
1328      1723  37537777477  CAD2      ADD       SP3  SF1               YES, IS %40;F2=BLANKMODE
1329      1724  37766361744            JMP  CAD7                  UNC   RETURN TO LOOP
1330      1725  01311210000  CAD3      SP1  POMI       SP1  010000      SET NEGATIVE SIGN
```

ADDRESS	CONTENTS	LABL	RBUS	SBUS	FUNC	SHFT	STOR	SPEC	SKIP	COMMENTS
1726	37777777677				ADD			CCL		
1727	25536777777	CAD4		SP3	INC		SP3			
1730	37766361743				JMP	CAD6			UNC	GOTO LOOP

```
1334
1335                                *
1336                                * IN THE LOOP PORTION OF THIS INSTRUCTION
1337                                * RA=SOURCE (ASCII) BYTE COUNT
1338                                * RC=TARGET (DECIMAL) DIGIT  COUNT
1339                                * RD=NON-ZERO COUNTER
1340                                * SP0=ABS WRD @ OF ASCII SOURCE
1341                                * SP1=THAT WHICH IS STORED
1342                                * SP2=ABS WRD @ OF BCD TARGET
1343                                * SP3=ASCII WORK WORD
1344                                * F1 = 'BLANKMODE'
1345                                * F2 = 'RIGHT BYTE'
1346                                *
1347          1731  33766121744     CAD5      RA   JMP  CAD7            POS      B RUNOUT? (SP3 SHOULD BE 0)
1348          1732  26537560437               OPND ADD  LR7  SP3  CF2  F2
1349          1733  17537764417               SBUS ADD  RR7  SP3  SF2  UNC      WAS NF2
1350          1734  37127377575          SP0       CAD       RSP0 ROD          WAS F2
1351          1735  25761010040               SP3  POMX            0040 NZRO   =%40?
1352          1736  37766361723                    JMP  CAD2            UNC      YES
1353          1737  37766143316                    JMP  TF14            F1       ILLEGAL IF EXPECTING BLANKS
1354          1740  25531537720               SP3  ROM       SP3  7720 NEG    SP3 <= %60 ?
1355          1741  16771537766               UBUS ROM            7766 NEG    (SP3 - %60) >= %12 ?
1356          1742  37766363316                    JMP  TF14            UNC      YES OR YES
1357          1743  25613377770     CAD6 RD   SP3  IOR       RD
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
1358          1744  37760773777     CAD7           OASR SR1                    SHIFT SP3:SP1 RIGHT 4
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
1359          1745  37760773777                    OASR SR1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
1360          1746  37760773277                    OASR SR1       INCT         INCR "4DIGITS?" CNTR
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
1361          1747  37760773777                    OASR SR1
1362          1750  37636537771          RC        INC       RC        NEG      INC A CNT(SBUS SHOULD BE 0)
1363          1751  37766361761                    JMP  CADP            UNC      JMP IF LAST BCD DIGIT
1364          1752  33676737777               RA   INC       RA        CTRM     4 DIGITS PROCESSED?INC BCNT
1365          1753  37766361731                    JMP  CAD5            UNC      NO,4 DIGITS NOT PROCESSED
1366          1754  35177777557               SP2  ADD       BUS  WRD
1367          1755  01177777437               SP1  ADD       BUS  DATA
1368          1756  35731777777               SP2  ROM       SP2  177777
1369          1757  37351777773                    ROM       CTRL 177773      CNTR _ -5
1370          1760  37306361731                    JMP  CAD5 SP1       UNC      CLEAR SP1 SO SBUS IN OASR=0
1371                                *
1372                                * HERE FOR LAST BCD WORD
1373                                *
1374          1761  14766031766     CAD8      CTRL JMP  *+5             ODD
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
1375          1762  37760773777                    OASR SR1                    SHIFT IN A 0
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
```

```
     ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP       COMMENTS                FRI, JUN  4, 1976, 10:14 AM

 1376           1763  37760773777                        OASR SR1
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
 1377           1764  37760773277                        OASR SP1        INCT
***  WARNING ( 4)  ***  SBUS MAY BE FORCED ON FOLLOWING LINE
***  WARNING (10)  ***  UBUS ON RBUS OR SR1 MISSING FROM OASR (TASR ON /20)
 1378           1765  37760773777                        OASR SP1
 1379           1766  37766331773                        JMP  CAD9              CTRM      4 DIGITS PROCESSED?
 1380           1767  35177777577                SP2     ADD        BUS  ROD               NO,GET LEFT BYTE
 1381           1770  01317770777                SP1     ADD  LRZ  SP1
 1382           1771  26777771777                UPND    ADD  LLZ
 1383           1772  01317777776          UBUS  SP1     ADD        SP1
 1384           1773  30777417777     CAD9       RD      ADD                       NZRO      ANY NON-ZERO DIGITS
 1385           1774  37777777737                        ADD                 CCE
 1386           1775  35177777557                SP2     ADD        BUS  WRD
 1387           1776  01177777437                SP1     ADD        BUS  DATA
 1388           1777  37766363302                        JMP  BPOP              UNC
```

```
1389
1390                                &2023
1391                                *
1392                                *THIS IS THE ENTRY POINT FOR NSLD AND SLD
1393          2023  37777777077     NSLD            ADD             SF3
1394          2024  23337777537     SLD       SM    ADD       SP0   CCRY
1395          2025  37777777766     SRLD X          ADD
1396          2026  16541600037           UBUS ROMN       X     000037          USE ONLY 5 LSB
1397          2027  23311600007     CKLN  SM    ROM       SP1   000007          SP1_@FINAL-SM+3
1398          2030  16767507762       Z   UBUS SUB                 CRRY
1399          2031  37571601752                ROM       PAR   BND2            EXIT TO STOV
1400          2032  33771507743           RA   ROM             7743 CRRY       >28?
1401          2033  31777517625         RBUS RC   ADD             CLO  NCRY     >28?
1402          2034  37766361452                JMP  TA17                UNC     LENGTH TRAP
1403          2035  33766002537           RA   JMP  ADZI            ZERO
1404          2036  31766002537           RC   JMP  ADZI            ZERO
1405                                *THIS SUBROUTINE DOES BOUNDS CHECKING FOR OPERAND B
1406                                *IT USES SP0 WHICH CONTAINS THE HIGHEST LEGAL ADDRESS IN THE STACK
1407                                *IT RETURNS THE WORD ADDRESS OF THE MSW IN SP2
1408                                *IT RETURNS THE WORD COUNT (0-7) IN OPND
1409                                *IT SETS F2 IF THE SIGN WORD IS A FULL WORD
1410          2037  01177777757     CKBR      SP1   ADD       BUS   WPS             @SM+7 (FINAL SM+3)
1411          2040  36177777437           PR    ADD       BUS   DATA
1412          2041  32777777377     CKB'      RR    ADD             LRF             F2_LSB(RR)
1413          2042  33777573377           RA    ADD   SR1       LRF   NF2         UBUS_BYTE COUNT - 1
1414          2043  16776777377           UBUS  INC             LRF               INCR UBUS IF LSB(RR)=1
1415          2044  16357773777           UBUS  ADD   SR1   CTRL                  CTRL _ WORD COUNT -1
1416          2045  32777773777           RR    ADD   SR1
1417          2046  22737777776         UBUS DR   ADD       SP2                 SP2_FIRST GUESS FOR WORD @
1418          2047  34767517776         UBUS DL   SUB                 NCRY      @MSW > DL ?
1419          2050  35767507775         SP0  SP2  SUB                 CRRY      SM > @MSW ?
1420          2051  35731700000               SP2  ROM       SP2   100000        TOGGLE IF NOT(RR<@MSW<SM)
1421          2052  34766777776         UBUS DL   BNDT
1422          2053  14777777765         RBUS CTRL ADD
1423          2054  16766707775         SP0  UBUS BND1                FSB
1424                                *
1425                                *
1426          2055  37777577766         X         ADD                 NF2
1427          2056  16771600002               UBUS ROM             000002        ADD 2 TO SHIFT IF B RT-JUST
1428          2057  16177777637               UBUS ADD       BUS   OPND          STORE IN OPND REG
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

1429
1430                                    *THIS SUBROUTINE CHECKS THE BOUNDS OF OPERAND A
1431                                    *IT ASSUMES SP0 CONTAINS THE HIGHEST LEGAL ADDRESS IN THE STACK
1432                                    *IT RETURNS THE WORD ADDRESS IN SP1
1433                                    *IT RETURNS THE WORD COUNT (0-7) IN SP3
1434                                    *IT SETS F2 IF THE SIGN WORD IS A FULL WORD
1435            2060  01176777757       CKAB       SP1   INC         BUS   WRS              @SM+8 (FINAL SM+4)
1436            2061  37177777422             Z          ADD         BUS   DATA
1437            2062  30777777377       CKA'       RD    ADD               LBF              F2_LSB(BYTE @)
1438            2063  31777573377                  RC    ADD   SP1         LBF   NF2        BYTE COUNT -1
1439            2064  16776777377                  UBUS  INC               LBF              INCR IF LSB(BYTE @)=1
1440            2065  16537773777                  UBUS  ADD   SP1   SP3                    SP3 _ WORD COUNT - 1
1441            2066  37167377754             SP1        CAD         BUS   WRS              @SM+6 (FINAL SM+2)
1442            2067  37177777426             X          ADD         BUS   DATA
1443            2070  30777773777                  RD    ADD   SP1
1444            2071  22317777776             UBUS DB    ADD         SP1                    SP1 _ FIRST GUESS AT WORD @
1445            2072  34767517776             UBUS DL    SUB                    NCRY       @MSW > DL ?
1446            2073  01767507775             SP0  SP1   SUB                    CRRY       SM > @MSW ?
1447            2074  01311700000                  SP1   ROM         SP1   100000           TOGGLE IF NOT(PB<@MSW<SM)
1448            2075  34766777776             UBUS DL    BNDT
1449            2076  25777777774             SP1  SP3   ADD
1450            2077  16766707775             SP0  UBUS  BNDT                   RSP
1451                                    *
1452                                    *
1453            2100  37757577766             X          ADD         PB         NF2        KEEP ORG X IN PB(NOW FREE)
1454            2101  26771777776                  OPND  ROM               177776           SHIFT -2 IF A RT-JUST.
1455            2102  16177667637                  UBUS  ADD         BUS   OPND NPRV
1456                                    *
1457                                    *
1458            2103  37762361346                        JSB   SPLT                 UNC
1459            2104  23176777741             SR   SM    INC         BUS   WRS
1460            2105  26177777437                  OPND  ADD         BUS   DATA
1461            2106  37762361331                        JSB   FREE                 UNC        MAYBE FREE (PL,DL,DB,Q)
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

1462
1463                                    *THIS SUBROUTINE TAKES PC AND RB AND RETURNS THE SIGN MASK
1464                                    *AND THE MSW MASK ( IN X AND Z )
1465            2107    37251607777     MSKA            POM         Z     007777          Z _'0FFF'
1466            2110    31635123777                 RC  CRS  SP1     RC          POS     SKIP IF EVEN # OF DIGITS
1467            2111    37251777777                 POM             Z     177777          Z _ 'FFFF'
1468            2112    23136777757                 SM  INC         BSP0  WRS             S-3
1469            2113    30177427437                 RD  ADD         BUS   DATA EVEN       SKIP IF EVEN BYTE ADDRESS
1470            2114    37257770762             Z       ADD   LRZ  Z                     Z _ '000F' OR '00FF'
1471            2115    37551600017                     POM         X     000017          X _ '000F'
1472            2116    30777437771             PC  RD  ADD                     ODD     RC HAS BYTE CNT-1 IN 15LSB
1473            2117    37551607400                     POM         X     007400          X _ '0F00'
1474            2120    37136777755             SP0     INC         PSP0  WRS             S-2
1475            2121    31175302437                 RC  CRS  SP1   BUS   DATA RSB
1476                                    *
1477                                    *
1478            2122    32775373317                 RB  CRS  SP1         HBF             F1 _ LSB OF PB FOR SCANB
1479            2123    36617777777                 PB  ADD         PD                    KEEP ORG X IN RD(NOW FREE)
1480                                    *THIS SUBROUTINE RETURNS THE SIGN AND MSW MASKS FOR A
1481                                    *IT EXPECTS SP0 TO CONTAIN S-1
1482                                    *IT RETURNS THE SIGN MASK IN PB AND THE MSW MASK IN RB
1483          2124  37751607777       MSKB            POM         PB    007777          PB _ '0FFF'
1484          2125  33675123777                 RA  CRS  SR1     RA          POS
1485          2126  37751777777                 POM             PB    177777          PB _ 'FFFF' IF RA IS EVEN
1486          2127  37136777755             SP0     INC         PSP0  WRS             S-1
1487          2130  32177427437                 RB  ADD         BUS   DATA EVEN       SKIP IF EVEN BYTE ADDRESS
1488          2131  36757770777                 PB  ADD   LRZ  PB                    PB _ '000F' OR '00FF'
1489          2132  37651600017                 POM             RB    000017          RB _ '000F'
1490          2133  32777437773             RA  RB  ADD                     ODD     RA HAS BYTE CNT-1 IN 15 LSB
1491          2134  37651607400                 POM             RB    007400          RB _ '0F00'
1492          2135  37136777755             SP0     INC         PSP0  WRS
1493          2136  33175372437                 RA  CRS  SR1     BUS   DATA
1494          2137  37477707055             SP0     ADD         SM    CLSR RSB
1495                                    *
1496                                    *
1497                                    *
1498                                    *
1499          2140  37766342232                     JMP  NSL'           F3          IF NSLD THEN JUMP
1500          2141  14722361507       SLDX      CTRL JSB  FTCH SP2     UNC
1501          2142  26757777777                 OPND ADD         PB
1502          2143  23176777777       CDG'      SM   INC         BUS   RDS             FETCH THE MODIFIED X
1503          2144  01326162151       CKDG      SP1  JMP  *+5   SP0         F2
1504          2145  21777077760       CKD' PL   0    DCAD                    NOFL
1505          2146  34766361456                 DL   JMP  TA15              UNC
1506          2147  22777077776             UBUS DR   DCAD                    NOFL
1507          2150  37766361456                 JMP  TA15              UNC
1508          2151  32777077773             RA  RB   DCAD                    NOFL
1509          2152  37766361456                 JMP  TA15              UNC
1510          2153  30777077771             RC  RD   DCAD                    NOFL
1511          2154  37766361456                 JMP  TA15              UNC
1512          2155  25737707777                 SP3  ADD         SP2         RSB
1513                                    *
1514                                    *
```

	ADDRESS	CONTENTS	LABL	RBUS	SBUS	FUNC	SHFT	STOR	SPEC	SKIP	COMMENTS
1515											
1516	2156	26341527774			OPND	ROMN		CTRL	7774	POS	DELETE 2LSB OF MOD. X
1517	2157	37766362227				JMP	SLSP			UNC	IF A RIGHT SHIFT IS NEEDED
1518	2160	26761410003			OPND	ROMN			0003	NZRO	
1519	2161	16766362173			UBUS	JMP	SLXY			UNC	
1520	2162	16761000001			UBUS	ROMX			0001	ZERO	
1521	2163	16766362166			UBUS	JMP	SLXW			UNC	
1522	2164	16762361024			UBUS	JSB	L1D'			UNC	
1523	2165	37766362173				JMP	SLXX			UNC	
1524	2166	16761000003	SLXW		UBUS	ROMX			0003	ZERO	11 NOT 10 CAUS PREV 01
1525	2167	37766362172				JMP	*+3			UNC	
1526	2170	37762361215				JSB	L2D'			UNC	
1527	2171	37777767777				ADD				UNC	
1528	2172	37762361120				JSB	L3D'			UNC	
1529	2173	14767773157	SLXX		CTRL	SUB	SP1		CTF		F1=1 IF #WD SHIFTS=0
1530	2174	16357773777			UBUS	ADD	SP1	CTRL			
1531	2175	35526142213			SP2	JMP	SLXZ	SP3		F1	JMP IF NO SHIFT NEEDED
1532	2176	33777407777	L4D'		RA	ADD				ZERO	WILL ANY DIGITS BE LOST?
1533	2177	37777777517				ADD			SCRY		IF SO, THEN SET CARRY
1534	2200	37766162206				JMP	SLXY			F2	JMP IF SHORT;UBUS_0
1535	2201	37437777777				ADD		0			
1536	2202	21457777777			0	ADD		DB			
1537	2203	22717777777			DB	ADD		DL			
1538	2204	34237777777			DL	ADD		PL			
1539	2205	37777777760		PL		ADD					
1540	2206	16617777777	SLXY		UBUS	ADD		RD			
1541	2207	30637777777			RD	ADD		RC			
1542	2210	31657777777			RC	ADD		RB			
1543	2211	32677737277			RB	ADD		RA	INCT	CTRM	SKIP WHEN DONE
1544	2212	37766362176				JMP	L4D'			UNC	
1545	2213	37317777455	SLXZ	SP0		ADD		SP1	CF1		REST @A IN SP1:CF1 FOR STOR
1546	2214	37722361557				JSB	STOR	SP2		UNC	
1547	2215	37777557777				ADD				NF1	STOR,F1=NON-STOR-DIG<>0
1548	2216	37777777517				ADD			SCRY		
1549	2217	26542171476	REST		OPND	JSB	RSTL	X		NF2	JSB IF LONG
1550	2220	00761410060	APOP		CTR	ROMN			0060	NZRO	
1551	2221	37777757777				ADD				NEXT	
1552	2222	23471777776			SM	ROM		SM	177776		
1553	2223	00761410040			CTR	ROMN			0040	NZRO	
1554	2224	37777757777				ADD				NEXT	
1555	2225	23471777776			SM	ROM		SM	177776		
1556	2226	37777757777				ADD				NEXT	

ADDRESS	CONTENTS	LABL	RBUS	SBUS	FUNC	SHFT	STOR	SPEC	SKIP	COMMENTS
2227	26762031064	SLSR		OPND	JSB	R1D			ODD	
2230	26762021246			OPND	JSB	R2D			EVEN	
2231	35526362213			SP2	JMP	SLXZ	SP3		UNC	
		*								
		*								
		*								
2232	37762361401	NSL'			JSB	SCNP			UNC	
2233	16666002141			UBUS	JMP	SLDX	PA		ZERO	SP0(UBUS) HAS SIG DIGITS
2234	23136777577			SM	INC		BSP0	ROD		FETCH SAVED MOD. X @ SM+1
2235	31635372777			RC	CRS	SL1	PC			RESTORE RC FROM MSKA
2236	33777777770		RD	RA	ADD					TARG LEN NEEDED (X+SIG)
2237	16767517771		RC	UBUS	SUB				NCRY	(ACT. TARG)>=(TARG NEEDED)?
2240	37766362141				JMP	SLDX			UNC	YES,JUST SLD
2241	33667507511		RC	RA	SUB		PA	SCRY	CRRY	NEW SHIFT AMOUNT
2242	37766361457				JMP	TA13			UNC	SET OVERFLOW IF SIG>TARG
2243	37176777755		SP0		INC		BUS	WRS		SM+2
2244	33167777430		RD	RA	SUB		BUS	DATA		REVISED X;UBUS_(ORGX-NEWX)
2245	26667777776		UBUS	OPND	SUB		BA			BA_(ORGX-NEWX)-(ORGX+-2)
2246	23176777757			SM	INC		BUS	WRS		STORE AT SM+1
2247	33167777437			RA	SUB		BUS	DATA		SM+1 _ NEWX+-2
2250	37766362141				JMP	SLDX			UNC	

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

1579
1580                                    *THIS IS THE SRD INSTRUCTION
1581            2251   23322362025      SRD        SM    JSB   SRLP SP0        UNC      CHECK B-BNDS,STOV,DIG&0-LEN
1582            2252   37777577766            X          ADD                    NF2      B-SIGN-WORD A HALF WORD?
1583            2253   16771777776               UBUS ROM                177776         NO DECR X BY 2
1584            2254   16177777637               UBUS ADD         BUS   OPND
1585            2255   37762362060                    JSB   CKAB                UNC      CHECK A BOUNDS
1586            2256   26771570002               OPND ROM                0002 NF2       A-SIGN WORD A HALF WORD?
1587            2257   16177777637               UBUS ADD         BUS   OPND           NO , INCR X BY 2
1588            2260   23176777741         SR    SM   INC         BUS   WRS
1589            2261   26177667437               OPND ADD         BUS   DATA SPRV
1590            2262   37762361346                    JSB   SPLT                UNC      SPLIT STACK?
1591            2263   37762361331                    JSB   FREF                UNC
1592            2264   37762362107                    JSB   MSKA                UNC
1593            2265   37762362124                    JSB   MSKB                UNC
1594            2266   14722361507               CTRL JSB   FTCH SP2            UNC
1595            2267   26742362143               OPND JSB   CDG' PB             UNC      CHECK FOR ILLEGAL DIGITS
1596            2270   26341527774               OPND ROMN        CTRL 7774 BUS  DELETE 2LSB OF MOD. X
1597            2271   37766362332                    JMP   SRSL                UNC      IF A LEFT SHIFT IS NEEDED
1598            2272   26761410003               OPND ROMN                0003 NZRO
1599            2273   16766362305               UBUS JMP   SRXX                UNC
1600            2274   16761000001               UBUS ROMX                0001 ZERO
1601            2275   16766362300               UBUS JMP   SRXW                UNC
1602            2276   16762361064               UBUS JSB   R1D                 UNC
1603            2277   37766362305                    JMP   SRXX                UNC
1604            2300   16761000003      SRXW      UBUS ROMX                0003 ZERO      11 NOT 10 CAUS PREV 01
1605            2301   37766362304                    JMP   *+3                 UNC
1606            2302   37762361246                    JSB   R2D                 UNC
1607            2303   37777767777                    ADD                       UNC
1608            2304   37762361155                    JSB   R3D                 UNC
1609            2305   14767773157      SRXX      CTRL SUB   SP1         CTF            F1=1 IF #WD SHIFTS=0
1610            2306   16357773777               UBUS ADD   SP1   CTRL
1611            2307   35526142321               SP2  JMP   SRXY SP3         F1       JMP IF NO SHIFT NEEDED
1612            2310   33657777777      R4D       RA   ADD         RB
1613            2311   32637777777      R4D'      RB   ADD         RC
1614            2312   31606162317               RC   JMP   *+5    RD         F2       DONE IF F2(SHORT)
1615            2313   30237777777               RD   ADD         RL
1616            2314   37717777760          RL        ADD         DL
1617            2315   34457777777               DL   ADD         DB
1618            2316   22437777777               DB   ADD         0
1619            2317   37677737277                    ADD         RA   INCT CTRM
1620            2320   37646362311                    JMP   R4D' RB          UNC
1621            2321   37777777766      SRXY X         ADD
1622            2322   16767057777               UBUS CAD                      BIT6     CLEAN UP LSW FOFF OR FFF0
1623            2323   16777771777               UBUS ADD   LLZ                         FOFF => F000
1624            2324   16777577777               UBUS ADD                      NF2
1625            2325   30603767776          UBUS RD   AND         RD          UNC      SHORT
1626            2326   21423777765          RBUS 0    AND         0                    LONG
1627            2327   37317777455          SP0       ADD         SP1   CF1            REST @A IN SP1;CF1 FOR STOR
1628            2330   37722361557                    JSB   STOR SP2          UNC
1629            2331   37766362217                    JMP   REST              UNC
1630            2332   26762031027      SRSL      OPND JSB   L1D                 ODD
1631            2333   26762021217               OPND JSB   L2D                 EVEN
1632            2334   35526362321               SP2  JMP   SRXY SP3         UNC
```

```
1633
1634                                * D4B CONVERTS 1 WORD (4 DIGITS) DEC TO 1 WORD BIN
1635                                * DEC WORD RECEIVED ON UBUS,SP1 - RETURNED IN SP3,UBUS
1636                                * SP2 IS USED FOR ALL 0 INDICATION (SP2=0)
1637                                * TRAPS ON INVALID DECIMAL
1638        2335  16777077777       D4B       UBUS DCAD                    NOFL
1639        2336  37766363767                       JMP  TCID              UNC       INVALID DEC TRAP
1640        2337  35733377774             SP1  SP2  IOR       SP2                    SP2 IS ZERO INDICATOR
1641        2340  01521770000                  SP1  ROMN      SP3  170000           EXTRACT MS DIGIT
1642        2341  16777773777                  UBUS ADD  SP1
1643        2342  16777773777                  UBUS ADD  SP1
1644        2343  25537773776             UBUS SP3  ADD  SP1  SP3
1645        2344  01761607400                  SP1  ROMN           007400           EXTRACT 2ND DIGIT
1646        2345  25537777776             UBUS SP3  ADD       SP3
1647        2346  16777773777                  UBUS ADD  SP1
1648        2347  16777773777                  UBUS ADD  SP1
1649        2350  25537773776             UBUS SP3  ADD  SP1  SP3
1650        2351  01761600360                  SP1  ROMN           000360           EXTRACT 3RD DIGIT
1651        2352  25537777776             UBUS SP3  ADD       SP3
1652        2353  16777773777                  UBUS ADD  SP1
1653        2354  16777773777                  UBUS ADD  SP1
1654        2355  25537773776             UBUS SP3  ADD  SP1  SP3
1655        2356  01761600017                  SP1  ROMN           000017           EXTRACT 4TH DIGIT
1656        2357  25537707776             UBUS SP3  ADD       SP3       RSB
```

	ADDRESS	CONTENTS	LABL	RBUS	SBUS	FUNC	SHFT	STOR	SPEC	SKIP	COMMENTS
1657											
1658			&2360								
1659	2360	37766362400				JMP	UNIM			UNC	
1660	2361	37766361275				JMP	DMPY			UNC	
1661	2362	37766361641				JMP	CVAD			UNC	
1662	2363	37766363323				JMP	CVDA			UNC	
1663	2364	37766363112				JMP	CVBD			UNC	
1664	2365	37766363450				JMP	CVDB			UNC	
1665	2366	37766362024				JMP	SID			UNC	
1666	2367	37766362023				JMP	NSLD			UNC	NSLD
1667	2370	37766362251				JMP	SPD			UNC	
1668	2371	37766362403				JMP	ADDD			UNC	
1669	2372	37766362402				JMP	SUBD			UNC	CMPD
1670	2373	37766362402				JMP	SUBD			UNC	
1671	2374	37766362566				JMP	MPYD			UNC	
1672	2375	37777777777				ADD					
1673	2376	37777777777				ADD					
1674	2377	37777777777				ADD					
1675	2400	37571607777	UNIM			ROM			PAR 007777		
1676			*								
1677			*								
1678			*								
1679	2401	37777777777				ADD					SPACER

```
1680
1681                                *CMPD AND SUBD
***  WARNING ( 2)  ***  RBR CONFLICTS WITH PREFETCH ON INSTR ENTRY
1682          2402  03156777017     SUBD      RBR  INC        SBR  ABS            SET ABS-BANK-REG TO 1
1683                                *ADDD
1684          2403  23322362027     ADDD      SM   JSB  CKLN SP0        UNC       CHECK STOV,ZLEN,B-BNDS
1685          2404  37762362060                JSB  CKAB             UNC       CHECK A-BNDS
1686          2405  24762131346           STA  JSB  SPLT             NEG
1687          2406  37762361327                JSB  FPAD             UNC
1688          2407  37762362107                JSB  MSKA             UNC
1689          2410  37762362124                JSB  MSKB             UNC
1690                                *THIS IS THE ADD SECTION
1691                                *UPON ENTRY IT EXPECTS:
1692                                *SP1 - ADDRESS OF MSW
1693                                *SP3 - WORD COUNT (0-7)
1694                                *X - SIGN MASK
1695                                *Z - MOST SIGNIFICANT WORD MASK
1696                                *F2 - "USE ONLY FOUR WORDS"
1697                                *F3 - "DON'T NEED TO SHIFT 'B'"
1698                                *IT USES SP0,F1,CNTR ( AND THEY ARE FREE AT THE END )
1699                                *IT RETURNS THE ANSWER IN (RA,RB,RC,RD) OR (RA,PB,RC,RD,PL,DL,DB,0)
1700                                *IT USES ABS-BANK-REG EVEN/ODD TO DETERMINE ACTUAL ADD/SUBTRACT
1701                                *UPON EXIT SP1 = @MSW,OPND = (MSW),F3 = CARRY OUT
1702          2411  14722361507           CTRL JSB  FTCH SP2        UNC
1703          2412  25117777574     ADSC SP1  SP3  ADD       RSP1 ROD            SP1 _ @LSW
1704          2413  26746342420           OPND JMP  *+5     PB        F3        JMP IF B DOESN'T NEED SHIFT
1705          2414  37777427766      X         ADD                    EVEN      IS MASK '0F00'?
1706          2415  37762361246                JSB  R2D               UNC       NO, SHIFT 'B' RIGHT
1707          2416  37777437766      X         ADD                    ODD       IS MASK '000F'?
1708          2417  37762361217                JSB  L2D               UNC       NO,SHIFT 'B' LEFT
1709          2420  37777777166      X         ADD            CF3
1710          2421  16767057777           UBUS CAD                    BIT6      SIGN '000F'?
1711          2422  16777771777           UBUS ADD  LLZ                         UBUS WAS 'F0FF';MAKE 'F000'
1712          2423  26163777636      UBUS OPND AND       BUS  OPND             OPND _ MASKED LSW
1713          2424  17763777666      X    SBUS AND            CCL              UBUS _ SIGN
1714          2425  16761410017           UBUS ROMN           0017 NZRO        IN POSITION '000X';YES-OK
1715          2426  17777776777           SBUS ADD  SWAP                       NO-ROTATE(ALSO IF ALL 0)
1716          2427  16761010015           UBUS ROMX           0015 NZRO        B-SIGN NEG?
1717          2430  03156767017           RBR  INC       SBR  ABS  UNC       INCR ABS-BANK-REG IF NEG
1718          2431  16777777717           UBUS ADD            CCG              SET CCG IF 'A' NOT NEG
1719          2432  16777427477           UBUS ADD            SF1  EVEN       SKIP IF ACTUAL ADDITION
1720          2433  37762361361                JSB  9CMP              UNC       COMPLEMENT,SF3 IF ACT. SUB
1721          2434  25347777777           SP3  CAD       CTRL             CNTR _ WORD COUNT
1722          2435  37327777777                CAD       SP0              SP0 _ %177777
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

1723
1724             2436   37766162463                JMP  ADSH                F2        JUMP IF SHORT(F2)
1725             2437   37762332535                JSB  ADDN                CTRM      JSB DONE IF CTRM
1726             2440   01117737575          SP0   SP1  ADD        BSP1 ROD  CTRM
1727             2441   26763777775          SP0   OPND AND
1728                                        *BEFORE CTRM (AND THE JSB TO DONE), SP0 = %177777
1729                                        *AND THE PREVIOUS TWO LINES WILL DECREMENT SP1 AND GIVE JUST OPND
1730                                        *IMMEDIATELY AFTER CTRM (AND THE JSB), SP0 = 0
1731                                        *AND THE PREVIOUS TWO LINES WILL PRODUCE A NEW REQUEST TO MSW
1732                                        *THE SP0(=0) OPND AND, SINCE IT IS NOP'ED, WILL BE 0 OPND(MODIFIED) ADD.
1733                                        *FOP LINES AFTER THAT, 0 OPND AND WILL PRODUCE 0.
1734             2442   21437077276          UBUS 0     DCAD       0     INCT NOFL
1735             2443   37766361456                JMP  TA15                UNC       JUMP ILLEGAL DIGIT TRAP
1736             2444   37762332535                JSB  ADDN                CTRM
1737             2445   01117737575          SP0   SP1  ADD        BSP1 ROD  CTRM
1738             2446   26763777775          SP0   OPND AND
1739             2447   22457077276          UBUS DB    DCAD       DB    INCT NOFL
1740             2450   37766361456                JMP  TA15                UNC
1741             2451   37762332535                JSB  ADDN                CTRM
1742             2452   01117737575          SP0   SP1  ADD        BSP1 ROD  CTRM
1743             2453   26763777775          SP0   OPND AND
1744             2454   34717077276          UBUS DL    DCAD       DL    INCT NOFL
1745             2455   37766361456                JMP  TA15                UNC
1746             2456   37762332535                JSB  ADDN                CTRM
1747             2457   01117737575          SP0   SP1  ADD        BSP1 ROD  CTRM
1748             2460   26763777775          SP0   OPND AND
1749             2461   16237077260          PL    UBUS DCAD       PL    INCT NOFL
1750             2462   37766361456                JMP  TA15                UNC
1751             2463   37762332535   ADSH         JSB  ADDN                CTRM
1752             2464   01117737575          SP0   SP1  ADD        BSP1 ROD  CTRM
1753             2465   26763777775          SP0   OPND AND
1754             2466   30617077276          UBUS RD    DCAD       RD    INCT NOFL
1755             2467   37766361456                JMP  TA15                UNC
1756             2470   37762332535                JSB  ADDN                CTRM
1757             2471   01117737575          SP0   SP1  ADD        BSP1 ROD  CTRM
1758             2472   26763777775          SP0   OPND AND
1759             2473   31637077276          UBUS RC    DCAD       RC    INCT NOFL
1760             2474   37766361456                JMP  TA15                UNC
1761             2475   37762332535                JSB  ADDN                CTRM
1762             2476   01117737575          SP0   SP1  ADD        BSP1 ROD  CTRM
1763             2477   26763777775          SP0   OPND AND
1764             2500   32657077276          UBUS RB    DCAD       RB    INCT NOFL
1765             2501   37766361456                JMP  TA15                UNC
1766             2502   37762332535                JSB  ADDN                CTRM
1767             2503   01117737575          SP0   SP1  ADD        BSP1 ROD  CTRM
1768             2504   26763777775          SP0   OPND AND
1769             2505   33677077776          UBUS RA    DCAD       RA    INCT NOFL
1770             2506   37766361456                JMP  TA15                UNC
```

```
1771
1772          2507  00766022541              CIR  JMP  CMP            EVEN          JMP IF COMPARE
1773                                   *THIS SECTION DETERMINES WHETHER OR NOT THE RESULT IN REGISTERS
1774                                   *HAS TO BE COMPLEMENTED
1775          2510  03777437017              RBR  ADD              ABS  ODD       SKIP IF ACTUAL SUBTRACT
1776          2511  37777747777                   ADD                   F3        SKIP IF CARRY OUT
1777          2512  37777777457                   ADD              CF1            HERE IF SUB OR ADD&NF3
1778                                   *                                          LEAVE F1 SET IF ALREADY OVF
1779          2513  37777747777                   ADD                   F3        SKIP IF CARRY OUT
1780          2514  03777437017              RBR  ADD              ABS  ODD       SKIP IF SUB
1781          2515  37766362531                   JMP  ADFN             UNC       HERE IF ADD OR SUB&F3
1782          2516  37762361361                   JSB  9CMP             UNC       HERE IF SUB&NF3(RECOMP)
1783          2517  24501200400              STA  POMX       STA  000400          CCL _ CCG OR CCG _ CCL
1784          2520  37766162525   10CM            JMP  *+5              F2
1785          2521  21437377777              Q    DCAD       Q
1786          2522  22457377777              DB   DCAD       DB
1787          2523  34717377777              DL   DCAD       DL
1788          2524  37237377760         PL        DCAD       PL
1789          2525  30617377777              RD   DCAD       RD
1790          2526  31637377777              RC   DCAD       RC
1791          2527  32657377777              RB   DCAD       RB
1792          2530  33677377777              RA   DCAD       RA
1793          2531  37722361557   ADFN            JSB  STOF SP2            UNC       STORE & TEST FOR OVF
1794          2532  26542171476              OPND JSB  RSTI Y              NF2       RESTORE PL,DL,DB,Q IF LONG
1795          2533  37766141453                   JMP  TX13             F1        JMP TRAP-ADD-OVERFLOW IF F1
1796          2534  37766362220                   JMP  APOP             UNC
1797                                   *THIS IS THE SUBROUTINE THAT IS CALLED BY THE ADDD "FETCH A"
1798                                   *PORTION OF THE INSTRUCTION FOR THE LAST 'A'WORD IN MEMORY
1799          2535  37337777777   ADDM            ADD       SP0
1800          2536  26163707622        7     OPND AND       BUS  OPND RSR
1801                                   *HERE FOR ZERO LENGTH OPERANDS OF ADDD FAMILY
1802          2537  37762361374   ADZL            JSB  PSHA             UNC
1803          2540  37766362220                   JMP  APOP             UNC
1804                                   *THIS DOES THE END OF THE COMPARE INSTRUCTION
***  WARNING ( 2)  ***  RBR CONFLICTS WITH PREFETCH ON INSTR ENTRY
1805          2541  03777437017   CMP      RBR  ADD              ABS  ODD       ACTUAL SUB ?
1806          2542  37777747637                   ADD              CLO  F3        NO,F3 ?
1807          2543  37777767637                   ADD              CLO  UNC       HERE IF SUB OR ADD&NF3
1808          2544  37766362564                   JMP  CPEN             UNC       DONE, ADJ. SM;ADD&F3;NO CCE
1809          2545  37777747777                   ADD                   F3
1810          2546  03777437017              RBR  ADD              ABS  ODD       NF3, SUB?
1811          2547  37777767777                   ADD                   UNC       F3&SUB,NF3&ADD
1812          2550  24501200400              STA  POMX       STA  000400          NF3&SUB;RECOMPLEMENT
1813          2551  31313377770         RD   RC   IOR        SP1
1814          2552  33773377772         RB   RA   IOR
1815          2553  01773007776         UBUS SP1  IOR                  ZERO
1816          2554  37766362564                   JMP  CPEN             UNC       NOT CCE,DONE
1817          2555  37777577777                   ADD                   NF2
1818          2556  37777777737                   ADD              CCE
1819          2557  22766162564              DB   JMP  CPEN             F2        DONE IF SHORT
1820          2560  21313377776         UBUS 0    IOR        SP1
1821          2561  34773377760         PL   DL   IOR
1822          2562  01773017776         UBUS SP1  IOR                  NZRO
1823          2563  37777777737                   ADD              CCE           SET CCE ON CMPD
```

ADDRESS	CONTENTS	LABL	RBUS	SBUS	FUNC	SHFT	STOR	SPEC	SKIP	COMMENTS
2564	37762361465	CPEM			JSB	RSTA			UNC	
2565	37766362220				JMP	APOP			UNC	

```
1826
1827                                *    MPYD DEC MULTIPLY
1828          2566  23322362027     MPYD      SM   JSB  CKLN SP0       UNC
1829                                * COMMON WITH NSLD - CHECKS VALID DIGIT # & ZERO DGIT #
1830                                * FOR BOTH A & B - BOUNDS FOR B & FOR STACK OVERFLOW
1831          2567  37762362060                JSB  CKAB            UNC       CHECK BOUNDS FOR B
1832          2570  32775263317           RB   CRS  SR1       HBF  NPRV      F1 SET IF @B ODD
1833          2571  37762361346                JSB  SPLT            UNC
1834          2572  37762361335                JSB  FRLG            UNC       FREE REGS LONG
1835          2573  37762362107                JSB  MSKA            UNC       MASK A
1836          2574  37762362124                JSB  MSKB            UNC       MASK B
1837                                * SAVE A PARAMS (MSW & SIGN MASK @W WORD #)
1838          2575  37771600011                ROM             000011         SM NOT YET SM' FROM MSKB
1839          2576  23137777756           UBUS SM   ADD       BSP0 WRS       SP0 = SM' + 9
1840          2577  37177777422           Z         ADD       BUS  DATA      MSW MASK
1841          2600  37777427426           X         ADD            CF2  EVEN  CLEAR F2 FOR TA13
1842          2601  25531700000                SP3  ROM       SP3  100000    SET MSB IF '000F'
1843          2602  37136777755           SP0       INC       BSP0 WRS
1844          2603  25177777437                SP3  ADD       BUS  DATA      WORD COUNT
1845          2604  37176777755           SP0       INC       BUS  WRS
1846          2605  01177777437                SP1  ADD       BUS  DATA      TARGET @
1847                                * SCAN B & A FOR LEADING ZEROS
1848                                * SCNB & SCNR RETURNS MODIFIED DIGIT # IN SP0
1849                                * ADDRESS,WORD #,MSW MASK ALSO MODIFIED
1850          2606  37437777766           X         ADD       C              SIGN MASK TO 0
1851          2607  37762361401                JSB  SCNB            UNC
1852          2610  16662361433           UBUS JSB  SCNA RA          UNC       RA_#DIGITS IN 'B'
1853                                * PICK SHORTER OPERAND - EXCHANGE IF PB(A)<PB(B)
1854          2611  16767137773           RA   UBUS CAD                 NEG       LEN(B) - LEN(A)
1855          2612  17626362757                SBUS JMP  DMPC PC          UNC       - EXCHANGE  (SLOW JUMP)
1856                                * CHECK DIGIT # < 15
1857          2613  33771537761                RA   ROM             7761 NEG       -15
1858          2614  37766361457                JMP  TA13            UNC       DIGIT # >14 OVERFL
1859                                * TRANSF A PARAM : X TO Q, Z KEPT, SP3 TO DB, SP1 TO DL
1860          2615  25457777777                SP3  ADD       DB             WORD COUNT
1861          2616  01717777417                SP1  ADD       DL   SF2       WORD @
1862                                * FETCH SHORTER OPERAND - CAN BE 5 WORDS
1863                                * SINCE F2 SET FTCH KEEPS MSW (DIGIT) IN DL
1864          2617  37551600017     DMP1       ROM       X    000017         FOR FTCH
1865          2620  14771537774           CTRL ROM             7774 NEG       -4  WORD # < 5
1866          2621  37766362771                JMP  DMP2            UNC            NO
1867          2622  14722361507           CTRL JSB  FTCH SP2         UNC
1868          2623  37777747777                ADD                  F3        SKIP IF LSW FULL
1869          2624  37762361155                JSB  R3D             UNC       LSW HALF
1870          2625  37762341064                JSB  R1D             F3        LSW FULL
1871          2626  37777777437                ADD             CF2            RESET TO LONG FOR OVRD TRAP
```

```
1872
1873                                    * CONVERT TO BINARY
1874          2627  37331623420 DMP3          POM         SP0  023420          10K FOR CVDB
1875          2630  33302362335               JSB  D4B    SP1       UNC       CONV MSW
1876          2631  16677777777               BUS  ADD    RA
1877          2632  32302362335               RB   JSB  D4B  SP1       UNC       CONV 2ND WORD
1878          2633  37762363745                    JSB  DBM1            UNC       COMB WORDS 1,2
1879          2634  31302362335               RC   JSB  D4B  SP1       UNC       CONV 3RD WORD
1880          2635  37762363737                    JSB  DBM2            UNC       COMB WORDS 1,2,3
1881          2636  30302362335               RD   JSB  D4B  SP1       UNC       CONV 4TH WORD
1882          2637  37762363731                    JSB  DBM3            UNC       COMPLETE CONV
1883                                    * BINARY MAX 3 WORDS (RB,RC,RD)
1884                                    * GET READY TO FETCH 2ND OPERAND
1885          2640  34737777777               DL   ADD        SP2                  ADDRESS
1886          2641  22357777777               DB   ADD        CTRL                 WORD COUNT
1887          2642  37757777762          Z         ADD        RB                   MSW MASK
1888          2643  21657777777               0    ADD        RB                   SIGN MASK
1889          2644  32537777777               RB   ADD        SP3                  SP3 _ MS BIN WORD
1890          2645  31317777777               RC   ADD        SP1                  SP1 _ MIDDLE BIN WORD
1891          2646  30257777777               RD   ADD        Z                    Z   _ LS BIN WORD
1892          2647  37551600017                    POM        X    000017          FOR FTCH
1893          2650  14722361507               CTRL JSB  FTCH SP2       UNC       FETCH 2ND OPERAND
1894                                    * SAVE LSW IN Z - NEXT WORD IN OPND - MSW ON STACK
1895                                    * AT SM+14 - SR SET TO INDICATE WORD # OF MPLER
1896                                    * SP=0 : 1, SP=1 : 2, SP=2 : 3
1897          2651  25546002655               SP3  JMP  *+4   X         ZERO      MSW = 0          (X   _ 0)
1898          2652  23176777757               SM   INC        BUS  WRS            MSW IF NOT ZERO
1899          2653  25177777437               SP3  ADD        BUS  DATA
1900          2654  37557767217                    ADD        X    INSR UNC       SR=1             (X   _ 0)
1901          2655  01766002660               SP1  JMP  *+3             ZERO      MIDDLE WORD 0 ?
1902          2656  01177777637               SP1  ADD        BUS  OPND
1903          2657  37777777217                    ADD                  INSR      SR=1 OR 2
1904          2660  37757747717                    ADD        PB   CCG  F3        SKIP IF LSW FULL(PB  _ 0)
1905          2661  37762361027                    JSB  L1D                 UNC       LEFT 1 DIGIT     (Q   _ 0)
1906          2662  37322341122                    JSB  L3D   SP0         F3        LEFT 3 DIGITS    (Q&SP0_0)
1907          2663  37522362145                    JSB  CKD'  SP3         UNC       CHECK VALID DEC  (SP2 _ 0)
1908                                    * SET CCL IF NEGATIVE (ABS ODD)
1909          2664  03777427017               RBR  ADD                  ABS  EVEN      SKIP IF POSITIVE
1910          2665  37777777677                    ADD                  CCL
1911                                    * CLEAR ACCUMULATOR FOR MPLY PB SP0 SP2 PL Z X Q
1912          2666  37537777442          Z         ADD        SP3  CF1            FIRST BINARY MULTIPLIER
1913          2667  37257777177                    ADD        Z    CF3                                 (Z   _ 0)
1914          2670  37317777760          PL        ADD        SP1                 5TH MS WORD OF MULTIPLICAND
1915          2671  37237777777                    ADD        PL                                      (PL  _ 0)
```

```
1916
1917                                * MULTIPLY
1918                                * DOUBLE        RA  RB  RC  RD  SP1 DL  DB
1919                                * (ACCUM)       PB  SP0 SP2 PL  Z   X   0
1920          2672  37351777757     DMP4                ROM         CTRL 177757          -17
1921          2673  25766022705                 SP3   JMP   DMP7              EVEN
1922          2674  21766341457     DMP5        0     JMP   TA13              F3        OVERFL >28 DIGITS
1923          2675  22437377776           UBUS  DB    DCAD        0
1924          2676  34557377766           X     DL    DCAD        X
1925          2677  01257377762           Z     SP1   DCAD        Z
1926          2700  30237377760           PL    RD    DCAD        PL
1927          2701  35737377771           RC    SP2   DCAD        SP2
1928          2702  32337377775           SP0   RB    DCAD        SP0
1929          2703  36757377773           RA    PB    DCAD        PB
1930          2704  37766341457     DMP6                JMP   TA13              F3        OVERFL >28 DIGITS
1931          2705  25537773777     DMP7        SP3   ADD   SP1   SP3
***  WARNING (12)  ***  ZERO,NZRO,NSME SKIP TESTS MADE ON T-BUS
1932          2706  16775013317                 UBUS  CRS   SP1         HBF   NZRO
1933          2707  37766202726                       JMP   DMP9              SRZ       JUMP IF DONE
1934          2710  22766332723                 DB    JMP   DMPP              CTRM      JUMP IF MORE WORDS NEEDED
1935          2711  22457377776           UBUS  DB    DCAD        DB
1936          2712  34777777277                 DL    ADD               INCT
1937          2713  34717377776           UBUS  DL    DCAD        DL
1938          2714  01317377774           SP1   SP1   DCAD        SP1
1939          2715  30617377770           RD    RD    DCAD        RD
1940          2716  31637377771           RC    RC    DCAD        RC
1941          2717  32657377772           RB    RB    DCAD        RB
1942          2720  33677157773           RA    RA    DCAL        RA        NF1
1943          2721  37766362674                       JMP   DMP5              UNC       ADD TO ACCUMULATOR
1944          2722  37766362704                       JMP   DMP6              UNC       ONLY DOUBLE
1945                                * TEST FOR MORE MULTIPLER WORDS
1946          2723  26537777237     DMP8        OPND  ADD         SP3   DCSR          2ND,3RD MPLER TO SP3
1947          2724  23176777777                 SM    INC         BUS   RDS
1948          2725  37766362672                       JMP   DMP4              UNC
```

```
ADDRESS   CONTENTS      LABL PBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

1949
1950                    * TRANSF PRODUCT TO STORE - READ BACK A
1951                    * PARAMETERS FOR STORE
1952      2726  23771600011 DMP9       SM    ROM             000011
1953      2727  16137777777           UBUS  ADD         PSP0  ROS         READ SIGN MASK
1954      2730  37637777775       SP0       ADD         PC
1955      2731  37457777766       X         ADD         PB
1956      2732  37717777762       %         ADD         DL
1957      2733  37677777777                 ADD         PA
1958      2734  37136777775       SP0       INC         PSP0  ROS         READ MSW MASK
1959      2735  26257777777             OPND  ADD         Z               MSW MASK
1960      2736  36657777777             PB    ADD         PB
1961      2737  35617777777             SP2   ADD         PD
1962      2740  37737777777                   ADD         SP2             CLEAR SP2 FOR STOR
1963      2741  37176777775       SP0       INC         PUS   ROS        READ ADDR
1964      2742  26357527317             OPND  ADD         CTRL  HBF  PDS  WD CNT TEMP. IN CTRL
1965      2743  37551430017                   ROM         X     0017 ODD  X _ '000F'
1966      2744  37551607400                   ROM         X     007400    X _ '0F00'
1967      2745  23771600003             SM    ROM               000003
1968      2746  16177777777             UBUS  ADD         PUS   ROS       RESTORE PB
1969      2747  26322151122             OPND  JSB   L3D   SP0         NF1 LSW HALF(L3D DOESN'T ALT.F1
1970      2750  37762141027                   JSB   L4D               F1  LSW FULL
1971      2751  37317777455       SP0       ADD         SP1   CF1         WORD @
1972      2752  14537777777             CTRL  ADD         SP3
1973      2753  26742361557             OPND  JSB   STOR  PB          UNC STORE PRODUCT
1974      2754  26542361476             OPND  JSB   RSTI  X           UNC RESTORE PL,DL,PB,O
1975      2755  37766141453                   JMP   TY13                F1  OVERFLOW
1976      2756  37766362220                   JMP   APOF                UNC COMPLETES INSTR
```

```
ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SPEC STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

1977
1978                                * THIS BRANCH EXCHANGES A & B FOR MPYD IF PB(A)<PB(B)
1979      2757  31771537761     DMP0       RC   ROM                 7761 NEG        RANK2 JUMP HERE - RC OK
1980      2760  37766361457                     JMP  TA13                UNC        TRAP PB>14
1981      2761  32437777417                RB   ADD        0     SP2              SIGN MASK
1982      2762  37657777766           X         ADD        PB
1983      2763  36257777777                PB   ADD        Z                      MSW MASK
1984      2764  37757777762           Z         ADD        PB
1985      2765  14457777777                CTRL ADD        PB                     WORD COUNT
1986      2766  25357777777                SP3  ADD        CTRL
1987      2767  35717777777                SP2  ADD        PL                     WORD @
1988      2770  01726362617                SP1  JMP  DMP1  SP2        UNC        BACK TO FETCH
1989                                * SPECIAL CASE OF FETCH SHORTER OPERAND IN MPYD (5 WORDS)
1990      2771  14722361507     DMP2       CTRL JSB  FTCH  SP2        UNC
1991      2772  33222361155                RA   JSB  P3D   PL         UNC        RIGHT JUST 4 LSW
1992      2773  36763777760           PL   PB   AND                             MS DIGIT
1993      2774  16777772776           UBUS UBUS ADD  SL1                        LEFT 2 BITS
1994      2775  16777772776           UBUS UBUS ADD  SL1                        2 MORE SHIFTS
1995      2776  16673377433           RA   UBUS IOR        RA    CF2            MS DIGIT COPIED
1996      2777  37766362627                     JMP  DMP3            UNC        CONTINUE
```

```
ADDRESS   CONTENTS     LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

1997
1998                                &3024
1999                                *THIS IS THE SECTION OF CODE THAT PERFOPMS THE CHAINNED DIVIDES
2000                                *BY 10,000 AND THE FAST DIVIDES BY 100 AND 10 TO GET 4 BCD DIGITS
2001                                *PER SUBROUTINE CALL
2002    3024  34317777777   D6          DL   ADD        SP1                          (SP2,DL)/10K
2003    3025  35772577437               SP2  REPN            CF2  21
2004    3026  36764332276         UBUS PB   DVSB SL1        INCT CTRM
2005    3027  37737773765         RBUS      ADD  SP1  SP2                          REM TO NEXT MSW
2006    3030  01717777777               SP1  ADD       DL                           STORE QUOTIENT IN DL
2007    3031  37317777760   D5    PL          ADD       SP1                          (SP2,PL)/10K
2008    3032  35772577437               SP2  REPN            CF2  21
2009    3033  36764332276         UBUS PB   DVSB SL1        INCT CTRM
2010    3034  37737773765         RBUS      ADD  SP1  SP2                          REM TO NEXT MSW
2011    3035  01237777777               SP1  ADD       PL                           STORE QUOTIENT IN PL
2012    3036  30317777777   D4          RD   ADD       SP1                          (SP2,RD)/10K
2013    3037  35772577437               SP2  REPN            CF2  21
2014    3040  36764332276         UBUS PB   DVSB SL1        INCT CTRM
2015    3041  37737773765         RBUS      ADD  SP1  SP2                          REM TO NEXT MSW
2016    3042  01617777777               SP1  ADD       RD                           STORE QUOTIENT IN RD
2017    3043  31317777777   D3          RC   ADD       SP1                          (SP2,RC)/10K
2018    3044  35772577437               SP2  REPN            CF2  21
2019    3045  36764332276         UBUS PB   DVSB SL1        INCT CTRM
2020    3046  37737773765         RBUS      ADD  SP1  SP2                          REM TO NEXT MSW
2021    3047  01637777777               SP1  ADD       RC                           STORE QUOTIENT IN RC
2022    3050  32317777777   D2          RB   ADD       SP1                          (SP2,RB)
2023    3051  35772577437               SP2  REPN            CF2  21
2024    3052  36764332276         UBUS PB   DVSB SL1        INCT CTRM
2025    3053  37737773765         RBUS      ADD  SP1  SP2                          REM TO NEXT MSW
2026    3054  01657777777               SP1  ADD       RB                           STORE QUOTIENT IN RB
2027    3055  33317777777   D1          RA   ADD       SP1                          (SP2,RA)/10K
2028    3056  35772577437               SP2  REPN            CF2  21
2029    3057  36764332276         UBUS PB   DVSB SL1        INCT CTRM
2030    3060  37737773765         RBUS      ADD  SP1  SP2
2031    3061  01677777777               SP1  ADD       RA                           STORE QUOTIENT IN RA
2032    3062  37751631000   D0                ROM       PB   031000                 100*256/2
2033    3063  37317777777                     ADD       SP1                          CLEAR SP1
2034    3064  35772707437               SP2  REPN            CF2  10
2035    3065  36764332276         UBUS PB   DVSB SL1        INCT CTRM
2036    3066  37737777765         RBUS      ADD       SP2                          GET (C*D00,00A*B)
2037    3067  37751650000                     ROM       PB   050000                 10*256*16/2
2038    3070  35772747437               SP2  REPN            CF2  04
2039    3071  36764332276         UBUS PB   DVSB SL1        INCT CTRM              GET (D000,0A*BC)
2040    3072  37317777765         RBUS      ADD       SP1
2041    3073  01737777777               SP1  ADD       SP2                          MAKE(0A*BC,D000)
2042    3074  37751602400                     ROM       PB   002400                 10*256/2
2043    3075  35772747437               SP2  REPN            CF2  04
2044    3076  36764332276         UBUS PB   DVSB SL1        INCT CTRM              GET (0BCD,000A)
2045    3077  37737777765         RBUS      ADD       SP2                          SP2 _ 0BCD
2046    3100  01777772774         SP1  SP1  ADD  SL1                               A * 4
2047    3101  16777772776         UBUS UBUS ADD  SL1                               00A0
2048    3102  16317776777               UBUS ADD  SWAP SP1                         SP1 _ A000
2049    3103  37751623420                     ROM       PB   023420                 10K
2050    3104  35737707774         SP1  SP2  ADD       SP2       PSR                SP2 _ ABCD
```

	ADDRESS	CONTENTS	LABL	RBUS	SBUS	FUNC	SHFT	STOR	SPEC	SKIP	COMMENTS
2051											
2052				*THIS SUBROUTINE ACCESSES AND COMPLEMENTS ANOTHER WORD							
2053				*F2 = "COMPLEMENT"							
2054				*F1 = "CARRY IN"							
2055	3105	37107167574	BDFT	SP1		CAD		BSP1	RCD	F2	FETCH NEXT WORD
2056	3106	26737707277			OPND	ADD		SP2	INCT	RSB	
2057	3107	16767147277			UBUS	CAD			INCT	F1	
2058	3110	16737707777			UBUS	ADD		SP2		RSB	
2059	3111	16736707157			UBUS	INC		SP2	CTF	RSB	

```
      ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

2060
2061          3112  23311600007  CVBD       SM   ROM         SP1  000007          FINAL SM + 3 FOR CKAB
2062          3113  16767507762        Z    UBUS SUB                      CRRY
2063          3114  37571601752             ROM         PAR  BND2              STOV
2064          3115  31771507743             RC   ROM              7743 CRRY
2065          3116  33771517771             RA   ROM              7771 CRRY
2066          3117  37766363310                  JMP  TRLN                  UNC
2067          3120  33766003306             RA   JMP  BDZL                  ZERO
2068          3121  31766003306             RC   JMP  BDZL                  ZERO
2069          3122  01177777757             SP1  ADD         BUS  WRS
2070          3123  36177777437             PB   ADD         BUS  DATA          SAVE PB AT SM'+3
2071          3124  22737777632       RB    DB   ADD         SP2  CLO           @MSW
2072          3125  34766777776       UBUS  DL   BNDT
2073          3126  23322362060             SM   JSB  CKAB   SP0           UNC
2074          3127  37762361335                  JSB  FRTC                  UNC
2075          3130  23337777777             SM   ADD         SP0
2076          3131  37767377773       RA         CAD                        WCNT - 1
2077          3132  35737777776       UBUS SP2  ADD         SP2            @LSW
2078          3133  16766777775       SP0  UBUS BNDT
2079          3134  22177777572       RB   DB   ADD         BUS  ROD       FETCH MSW
2080          3135  37237667717                  ADD         PL   CCG  MPRV  ZERO PL FOR CB1,2W
2081          3136  37762361346                  JSB  SPLT                  UNC   SPLIT STACK?
2082          3137  35722362107             SP2  JSB  MSKA   SP2           UNC   RENEW SP2 IF SPLT MODIFIED
2083          3140  26775022377             OPND CPS  SI1           LBF  EVEN  F2="NEGATIVE"
2084          3141  37777777677                  ADD                 CCL         SET NEGATIVE SIGN
2085          3142  37136777755       SP0        INC         BSP0 WRS
2086          3143  32177777437             RB   ADD         BUS  DATA
2087          3144  37717777777                  ADD         PL                  ZERO DL FOR BD1W
2088          3145  37476777055       SP0        INC         SM   CLSP
2089          3146  37136777755       SP0        INC         BSP0 WRS
2090          3147  33177777437             RA   ADD         BUS  DATA
2091          3150  01337777477             SP1  ADD         SP0  SF1           SAVE @A IN SP0
2092          3151  37751623420                  ROM         FB   023420
2093          3152  35117777577             SP2  ADD         FSP1 ROD           FETCH LSW
2094          3153  33347377777             RA   CAD         CTRL               CTRL = - (WD CNT +1)
2095          3154  37602363105                  JSB  BDFT   RD            UNC   ZERO RD FOR BD1,2,3W
2096          3155  16666333260             UBUS JMP  BD1W   RA            CTRM  FIX RA;JMP IF CNT=1
2097          3156  37762363105                  JSB  BDFT                  UNC
2098          3157  16646333251             UBUS JMP  BD2W   PB            CTRM  FIX PB;JMP IF CNT=2
2099          3160  37762363105                  JSB  BDFT                  UNC
2100          3161  16626333240             UBUS JMP  BD3W   PC            CTRM  JMP IF CNT=3
2101          3162  37762363105                  JSB  BDFT                  UNC
2102          3163  16606333225             UBUS JMP  BD4W   PD            CTRM  JMP IF CNT=4
2103          3164  37762363105                  JSB  BDFT                  UNC
2104          3165  16226333206             UBUS JMP  BD5W   PL            CTRM
2105          3166  37762363105                  JSB  BDFT                  UNC
2106          3167  16706003206             UBUS JMP  BD5W   DL            ZERO
2107          3170  37722363024                  JSB  D6     SP2           UNC   0,DL,PL,RD,RC,RB,RA/10K
2108          3171  16437777777             UBUS ADD         0                  FIRST RESULT WORD
2109          3172  34722363031             DL   JSB  D5     SP2           UNC   DL,PL,RD,RC,RB,RA/10K
2110          3173  37737777760       PL         ADD         SP2                PL,RD,RC,RB,RA/10K
2111          3174  35442363036             SP2  JSB  D4     DB            UNC   SECOND RESULT WORD
2112          3175  16717777777             UBUS ADD         DL                 THIRD RESULT WORD
2113          3176  30722363043             RD   JSB  D3     SP2           UNC   RD,RC,RB,RA/10K
```

ADDRESS	CONTENTS	LABL	PBUS	SBUS	FUNC	SHFT	STOR	SPEC	SKIP	COMMENTS
2114	3177	16237777777		UBUS	ADD		PL			FOURTH RESULT WORD
2115	3200	31722363050		RC	JSB	D2	SP2		UNC	RC,RB,RA/10K
2116	3201	16617777777		UBUS	ADD		RD			FIFTH RESULT WORD
2117	3202	32722363055		RB	JSB	D1	SP2		UNC	RB,RA/10K
2118	3203	16637777777		UBUS	ADD		RC			SIXTH RESULT WORD
2119	3204	37722363055			JSB	D1	SP2		UNC	0,RA/10K
2120	3205	16646363266		UBUS	JMP	BDEN	RB		UNC	7TH RESULT WORD;RA HAS 8TH
2121	3206	37777417760	BD5W	PL		ADD			NZRO	
2122	3207	37766363225				JMP	BD4W		UNC	JMP IF PL=0
2123	3210	37722363031				JSB	D5	SP2	UNC	0,PL,RD,RC,RB,RA/10K
2124	3211	37737777760		PL		ADD		SP2		PL,RD,RC,RB,RA/10K
2125	3212	35422363036		SP2	JSB	D4	0		UNC	FIRST RESULT WORD
2126	3213	16457777777		UBUS	ADD		RB			SECOND RESULT WORD
2127	3214	30722363043		RD	JSB	D3	SP2		UNC	RD,RC,RB,RA/10K
2128	3215	16717777777		UBUS	ADD		RL			THIRD RESULT WORD
2129	3216	31722363050		RC	JSB	D2	SP2		UNC	RC,RB,RA/10K
2130	3217	16237777777		UBUS	ADD		PL			FOURTH RESULT WORD
2131	3220	32722363055		RB	JSB	D1	SP2		UNC	RB,RA/10K
2132	3221	16617777777		UBUS	ADD		RD			FIFTH RESULT WORD
2133	3222	37722363055			JSB	D1	SP2		UNC	0,RA/10K
2134	3223	16637777777		UBUS	ADD		RC			6TH RESULT WORD;RA HAS 7TH
2135	3224	33646363265		RA	JMP	BDEA	RB		UNC	25TH DIGIT TO RB
2136	3225	30766003240	BD4W	RD	JMP	BD3W			ZERO	JMP IF RD=0
2137	3226	37722363036				JSB	D4	SP2	UNC	0,RD,RC,RB,RA/10K
2138	3227	16437777777		UBUS	ADD		0			FIRST RESULT WORD
2139	3230	30722363043		RD	JSB	D3	SP2		UNC	
2140	3231	16457777777		UBUS	ADD		RB			SECOND RESULT WORD
2141	3232	31722363050		RC	JSB	D2	SP2		UNC	RC,RB,RA/10K
2142	3233	16717777777		UBUS	ADD		RL			THIRD RESULT WORD
2143	3234	32722363055		RB	JSB	D1	SP2		UNC	RB,RA/10K
2144	3235	16237777777		UBUS	ADD		PL			FOURTH RESULT WORD
2145	3236	33722363062		RA	JSB	D0	SP2		UNC	GET 4 DIGITS FROM RA
2146	3237	16606363263		UBUS	JMP	BDEC	RD		UNC	FIFTH RESULT WORD
2147	3240	31766003251	BD3W	RC	JMP	BD2W			ZERO	JMP IF RC=0
2148	3241	37722363043				JSB	D3	SP2	UNC	0,RC,RB,RA/10K
2149	3242	16437777777		UBUS	ADD		0			FIRST RESULT WORD
2150	3243	31722363050		RC	JSB	D2	SP2		UNC	RC,RB,RA/10K
2151	3244	16457777777		UBUS	ADD		RB			SECOND RESULT WORD
2152	3245	32722363055		RB	JSB	D1	SP2		UNC	RB,RA/10K
2153	3246	16717777777		UBUS	ADD		RL			THIRD RESULT WORD
2154	3247	33722363062		RA	JSB	D0	SP2		UNC	GET 3 DIGITS FROM RA
2155	3250	16226363263		UBUS	JMP	BDEC	PL		UNC	FOURTH RESULT WORD
2156	3251	32766003260	BD2W	RB	JMP	BD1W			ZERO	JMP IF RB=0
2157	3252	37722363050				JSB	D2	SP2	UNC	0,RB,RA/10K
2158	3253	16437777777		UBUS	ADD		0			FIRST RESULT WORD
2159	3254	32722363055		RB	JSB	D1	SP2		UNC	RB,RA/10K
2160	3255	16457777777		UBUS	ADD		RB			SECOND RESULT WORD
2161	3256	33722363062		RA	JSB	D0	SP2		UNC	GET 2 DIGITS FROM RA
2162	3257	16706363263		UBUS	JMP	BDEC	RL		UNC	THIRD RESULT WORD
2163	3260	37722363055	BD1W			JSB	D1	SP2	UNC	0,RA/10K
2164	3261	16437777777		UBUS	ADD		0			FIRST RESULT WORD
2165	3262	33457777777		RA	ADD		RB			GET ONE DIGIT FROM RA
2166	3263	37637777777	BDEC			ADD		RC		CLEAR RC
2167	3264	37657777777				ADD		RB		CLEAR RB
2168	3265	37677777777	BDEA			ADD		RA		CLEAR RA

```
                ADDRESS    CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

2169             3266   23771600003     BDEN       SM   ROM                000003
2170             3267   16177777577                UBUS ADD        PUS   ROD          GET PB
2171             3270   37777427426           X         ADD              CF2  EVEN     IS X = '0F00';SET LONG
2172             3271   25722361027                SP3  JSB  L1D   SP2        UNC      NO, IS '000F'
2173             3272   37777437446           X         ADD              CF1  ODD      IS X = '000F'?
2174             3273   25722361122                SP3  JSB  L3D   SP2        UNC      NO, IS '0F00'
2175             3274   37317777775           SP0       ADD              SP1           RESTORE @A
2176             3275   35537777777                SP2  ADD              SP3           RESTORE WD-CNT-A
2177             3276   26757777777                OPND ADD              PB
2178             3277   37722361557                     JSB  STOP  SP2        UNC
2179             3300   26542361476                OPND JSB  RSTI  X          UNC
2180             3301   37766143315                     JMP  TB13             F1       JUMP IF OVERFLOW
2181             3302   00761400020     BPOP       CIR  ROMN                0020 ZERO
2182             3303   23771777776                SM   ROM                177776
2183             3304   16471777776                UBUS ROM        SM    177776
2184             3305   37777757777                     ADD                   NEXT
2185             3306   37762361374     BDZL             JSB  PSHA             UNC
2186             3307   37766363302                      JMP  BPOP             UNC
2187             3310   37762361374     TBLN             JSB  PSHA             UNC
2188             3311   37531600016                      ROM             SP3  000016
2189             3312   33771507771                RA    ROM                7771 CRRY
2190             3313   37531600017     TE17             ROM             SP3  000017
2191             3314   37766363317                      JMP  TBUT             UNC
2192             3315   37531520013     TB13             ROM             SP3  0013 POS    EVERYTHING ALREADY RESTORED
2193             3316   37531600014     TE14             ROM             SP3  000014
2194             3317   24777777777     TBUT       STA   ADD
2195             3320   16777522616           UBUS UBUS ADD  SL1              SOV  POS
2196             3321   37571603134                     ROM             EAR  TRP0
2197             3322   37766363302                      JMP  BPOP             UNC
```

```
      ADDRESS    CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

2198
2199          3323  23311600007  CVDA       SM   ROM         SP1  000007          SET SP1 FOR CKA'
2200          3324  16767507762        Z    UBUS SUB                       CRRY
2201          3325  37571601752             ROM         RAR  BND2
2202          3326  23322361374             SM   JSR  PSHA SP0         UNC      RA:SAD;RB:TCNT;RC:TAD
2203          3327  32771517743             RB   ROM              7743 NCRY     >28
2204          3330  37766363442             JMP  TC17                  UNC      >28
2205          3331  37336777635        SP0      INC         SP0  CLO           SP0 _ S-3 FOR BOUNDS TSTS
2206          3332  37767377232        RB       CAD              DCSR          DEC. SR FOR FOLLOWING PUSH
2207          3333  32657777776        UBUS RB  ADD         RB                FIX A DIGIT CNT FOR COVS
2208          3334  17206003437             SBUS JMP  CFOF PUSH         ZERO     JUST SDEC IF OLD RB=0
2209          3335  37762362062             JSR  CKA'                  UNC      A:DCNT;B:SAD;C:DCNT*2;D:TAD
2210          3336  24762131346             STA  JSP  SPLT                  NEG
2211          3337  01177777577             SP1  ADD         BUS  ROD          FETCH FIRST TARGET WORD
2212          3340  37522362041             JSR  CKB' SP3              UNC      CLEAR SP3;CHECK B-BNDS
2213          3341  30777667377             RD   ADD                   LBF  MPRV F2="FIRST ASCII IN RIGHT"
2214          3342  35762361346             SP2  JSR  SPLT                  UNC      FIX SP2 IF SPLIT STACK
2215          3343  16137577577             UBUS ADD         PSP0 ROD     NF2      FETCH 1ST;SP2 OK EITHER WAY
2216          3344  26537771777             OPND ADD  LLZ    SP3                   SP3 _ GARBAGE BYTE IF F2
2217          3345  37307377774        SP1      CAD         SP1                   DECR SP1 FOR LATER INCR
2218          3346  37351777773             ROM         CTRL 177773                CNTR _ -5
2219          3347  37611777766             ROM         RD   177766                -10  USED FOR VALIDITY
2220          3350  32777427777             RB   ADD                        EVEN     IS SAD EVEN?
2221          3351  26737765457             OPND ADD  RLZ    SP2  CF1  UNC
2222          3352  26737767457             OPND ADD         SP2  CF1  UNC          SP2_ @'ED BCD BYTE IN LEFT
2223          3353  37351777775             ROM         CTRL 177775                CNTR _ -3 IF RA ODD
2224          3354  37136777575        SP0      INC         RSP0 ROD              GET NEXT SOURCE WORD
2225          3355  37657777777             ADD         RB                    CLEAR RB(NON-ZERO FLAG)
2226          3356  33766033361             RA   JMP  DA1                  ODD      NOTE:IF JMP;UBUS=SP2;SL4
2227          3357  35737776277             SP2  ADD  SHAR SP2  INCT
2228          3360  16766363365             UBUS JMP  DA2                  UNC      NOTE:IF JMP;UBUS=SP2;SL8
```

```
                ADDRESS   CONTENTS       LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS
2229
2230                                     * IN THE MAIN PORTION OF THIS INSTRUCTION
2231                                     * RA = DIGIT COUNT
2232                                     * RB = 'NON-ZERO DIGIT STORED' - (NOT CCE)
2233                                     * RC = SELECTED BCD DIGIT
2234                                     * RD = %177766 (-10)
2235                                     * SP0 = @BCD
2236                                     * SP1 = @ASCII
2237                                     * SP2 = BCD WORD (ROTATES)
2238                                     * SP3 = THAT WHICH IS STORED
2239                                     * F2 = 'RIGHT ASCII BYTE'
2240                                     * F1 = 'ILLEGAL DIGIT'
2241          3361  16775372777  DA1          UBUS CRS  SI1
2242          3362  16775372777               UBUS CRS  SI1
2243          3363  16775372777               UBUS CRS  SI1
2244          3364  16735372777               UBUS CRS  SI1  SP2
2245          3365  16621400017  DA2          UBUS ROMN       RC   0017 ZERO
2246          3366  16657777150        RD    UBUS ADD        RB   CTF            THIS LINE IFF DIG<>0
2247          3367  25531600060               SP3  ROM        SP3  000060
2248          3370  37766143443                    JMP  IC15                F1
2249          3371  37667017273        RA          CAD        RA   INCT NZRO      DIGIT CNT - 1 = 0 ?
2250          3372  37766363405                    JMP  DA4                 UNC      PROCESS SIGN
2251          3373  25533167411        RC    SP3  IOR        SP3  SF2  F2        RIGHT ASCII BYTE ?
2252          3374  16766363400               UBUS JMP  DA3                 UNC      NO, SKIP NEXT SECTION
2253          3375  01116777557               SP1  INC        RSP1 WRD              WRITE IT
2254          3376  25177777437               SP3  ADD        BUS  DATA
2255          3377  37537777437                    ADD        SP3  CF2              CLEAR SP3 FOR NEXT PASS
2256          3400  16537736777  DA3          UBUS ADD  SWAP  SP3            CTRM     SOURCE WORD FINISHED ?
2257          3401  35766363361               SP2  JMP  DA1                 UNC      NO,GO BACK
2258          3402  37351777773                    ROM        CTRL 177773           CNTR = -5
2259          3403  37136777575        SP0         INC        RSP0 ROD
2260          3404  26726363361               OPND JMP  DA1  SP2            UNC      LOOP
```

```
2261
2262                                *
2263                                * HERE TO PROCESS SIGN
2264                                *
2265          3405  35737770717     DA4         SP2   ADD  LRZ  SP2   CCG          SIGN DIGIT TO RT BYTE
2266          3406  01176777577                 SP1   INC        BUS   RDD          PREFETCH LAST TARG WORD
2267          3407  00761410100                 CIR   ROMM             0100 NZRO    BIT 9 OF CIR
2268          3410  35731047417                 SP2   ROMI       SP2   7417 MSME    ABS SIGN ? (00FX?)
2269          3411  25766363426                 SP3   JMP  DA6              UNC      1 OR 2 OVHD JMP;UBUS=SP3
2270          3412  25777777677                 SP3   ADD              CCL
2271          3413  25531600031                 SP3   ROM        SP3   000031       SP3 _ X,%111 (MAKES NEG)
2272          3414  35771440040                 SP2   ROM              0040 MSME    NEG SIGN? (SP2=%177737?)
2273          3415  37766363423                 JMP   DA5              UNC      SKIP NEXT SECT. IF NEG SIGN
2274          3416  25531777747                 SP3   ROM        SP3   177747       RESTORE %60(ABS)
2275          3417  37777777717                 ADD              CCG
2276          3420  00761400040                 CIR   ROMM             0040 ZERO    BIT 10 OF CIR
2277          3421  37766363426                 JMP   DA6              UNC      ASSUME ABS IF NOT NEG;U=SP3
2278          3422  25531600020                 SP3   ROM        SP3   000020       SP3 _ X,%100 (MAKES POS)
2279          3423  31766013426     DA5         RC    JMP  DA6              NZRO     1<=RC<=9;IF JMP,UBUS=SP3
2280          3424  25531430064                 SP3   ROM        SP3   0064 ODD     MAKE SP3=X,%164(+)OR%175(-)
2281          3425  16771600007                 UBUS  ROM              000007       IF +%164(+),MAKE %173
2282          3426  16537777771     DA6   RC    UBUS  ADD        SP3
2283          3427  37766163433                 JMP   DA7              F2       IF RIGHT BYTE,JUST STORE
2284          3430  25537775777                 SP3   ADD  PLZ  SP3
2285          3431  26777774777                 OPND  ADD  PRZ
2286          3432  25537777776           UBUS  SP3   ADD        SP3
2287          3433  01176777557     DA7         SP1   INC        BUS   WRD
2288          3434  25177777437                 SP3   ADD        BUS   DATA
2289          3435  32777417777                 RB    ADD              NZRO     ANY NON-ZERO DIGITS?
2290          3436  37777777737                 ADD              CCE
2291          3437  00761400020     CPOP        CIR   ROMM             0020 ZERO
2292          3440  23771777776                 SM    ROM              177776       SDEC BY 2
2293          3441  37467357776           UBUS        CAD        SM        NEXT     SDEC BY 1 MORE
2294          3442  37531520017     TC17        TOM              SP3   0017 POS
2295          3443  37531600015     TC15        ROM              SP3   000015
2296          3444  24777777617                 STA   ADD              SGV
2297          3445  16777522776           UBUS  UBUS  ADD  SI1              POS
2298          3446  37571603134                 ROM        RAE   TRP0
2299          3447  37766363437                 JMP   CPOP             UNC
```

```
2300
2301                                  * CONV DEC TO BIN
2302          3450  23311600007   CVDB      SM    POM         SP1   000007          FINAL SM + 3
2303          3451  16767507762         Z   UBUS  SUB                     CRRY
2304          3452  37571601752               POM         PAR   BND2              STOV
2305          3453  33771517743               RA    POM               7743 NCRY    DIG # > 28
2306          3454  37766363764               JMP   TD17                    UNC    DIG # TRAP
2307          3455  33766003776               RA    JMP   DBZI                ZERO
2308          3456  23136777757               SM    INC         RSP0  WRS           SAVE RD
2309          3457  30177777437               RD    ADD         BUS   DATA          -"-
2310          3460  22537777631         RC    DB    ADD         SP3   CLO           TARGET @
2311          3461  34766777776         UBUS  DL    BNDT                            LOW BOUND
2312          3462  37762362037               JSR   CKBP                    UNC    COMMON TESTS
2313          3463  37762363753               JSR   DRWC                    UNC    RETURNS @LSW IN SP1
2314          3464  01766777775         SP0   SP1   BNDT                            HIGH BOUND ;SLOW RSB
2315          3465  37136777755         SP0         INC         RSP0  WPS           WRITE S-2
2316          3466  31177667437               RC    ADD         BUS   DATA  NPRV     SAVE RC
2317          3467  37762361346               JSR   SPLT                    UNC
2318          3470  37762362124               JSR   MSKB                    UNC    GET MASKS
2319          3471  32302361332               RB    JSR   FREE  SP1               UNC    FREE REGS SHORT/LONG
2320          3472  14722361507               CTRL  JSR   FTCH  SP2               UNC    GET SOURCE
2321          3473  23177777777               SM    ADD         BUS   ROS           DIG # TO OPND TO CTRL
2322          3474  26757777777               OPND  ADD         PB                  RESTORE PB
2323          3475  01775373317               SP1   CRS   SP1         HRF           F1="SIGN MASK IS 000F"
2324          3476  25337557177               SP3   ADD         SP0   CF3   NF1     (CF3 FOR DCAD)SP0_@A
2325          3477  37762361064                     JSR   R1D                     UNC    1 DIG TO RIGHT(WON'T ALT.F1
2326          3500  37177777635         SP0         ADD         BUS   OPND          OPND GETS WORD @
2327          3501  26342151155               OPND  JSR   R3D   CTRL        NF1     3 DIG TO RIGHT(MAY ALT. F1)
2328          3502  37331623420                     POM         SP0   023420        10K FOR CONV
2329                                * NEXT BRANCH ACCORDING TO TARGET W-CNT
2330          3503  14771527773               CTRL  ROM                 7773 POS      -5
2331          3504  37726363701                     JMP   DBW1  SP2               UNC    DIG #<5 1 WORD
2332          3505  14771527766               CTRL  ROM                 7766 POS      -10
2333          3506  32726363661               RB    JMP   DBW2  SP2               UNC    DIG#<10 2 WORDS
2334          3507  14771527757               CTRL  ROM                 7757 POS      -17
2335          3510  37726363635                     JMP   DB4S  SP2               UNC    DIG #<17 4 WORD SHORT
2336          3511  14771527755               CTRL  ROM                 7755 POS      -19
2337          3512  37726363574                     JMP   DB4L  SP2               UNC    DIG #<19 4WORDS LONG
2338                                *  Z & X- NEVER CHANGED
2339                                *  PB   - RESTORED
2340                                *  SP0  - 10K
2341                                *  SP1  - DB4
2342                                *  SP2  - NON CCE
2343                                *  SP3  - DB4
2344                                *  OPND - WORD ADDRESS
```

```
2345
2346                                * 6 WORD CASE NEXT
2347          3513  37737777777             ADD        SP2                    CLEAR SP2 FOR 6 WD CASE
2348          3514  32302362335       RB    JSB  D4B   SP1        UNC       CONV MS 4 DIGITS
2349          3515  16677777777       UBUS  ADD        RA                    4 DIGITS CONVERTED
2350          3516  31302362335       RC    JSB  D4B   SP1        UNC       CONV NEXT 4 DIGITS
2351          3517  37762363745             JSB  DBM1              UNC       COMB WORDS 1,2
2352          3520  30302362335       RD    JSB  D4B   SP1        UNC       CONV 3RD WORD
2353          3521  37762363737             JSP  DBM2              UNC       COMB WORDS 1,2,3
2354          3522  37777777760   PL        ADD
2355          3523  16302362335       UBUS  JSB  D4B   SP1        UNC       CONV 4TH WORD
2356          3524  37762363731             JSB  DBM3              UNC       COMB WORDS 1,2,3,4
2357          3525  34302362335       DL    JSB  D4B   SP1        UNC       CONV 5TH WORD
2358          3526  37762363723             JSB  DBM4              UNC       COMB WORDS 1,2,3,4,5
2359          3527  22302362335       DB    JSB  D4B   SP1        UNC       CONV 6TH WORD
2360          3530  37762363715             JSB  DBM5              UNC       COMB WORDS 1,2,3,4,5,6
2361          3531  21302362335       0     JSB  D4B   SP1        UNC       CONV 7TH WORD
2362          3532  37762363707             JSB  DBM6              UNC       COMPLETE CONVERSION
2363          3533  03777437017       RBR   ADD                ABS  ODD       IF SIGN - ABS=1
2364          3534  37766363551             JMP  DB6A              UNC       POSITIVE SKIP COMPLEMENT
2365          3535  22447507777       DB    SUB        DB         CRRY      COMPLEMENT
2366          3536  34707367457       DL    CAD        DL    CF1  UNC
2367          3537  34707777157       DL    SUB        DL    CTF
2368          3540  37777547760   PL        ADD                   F1
2369          3541  16227367777       UBUS  CAD        PL         UNC
2370          3542  16227507777       UBUS  SUB        PL         CRRY
2371          3543  30607367777       RD    CAD        RD         UNC
2372          3544  30607507777       RD    SUB        RD         CRRY
2373          3545  31627367777       RC    CAD        RC         UNC
2374          3546  31627507777       RC    SUB        RC         CRRY
2375          3547  32647367777       RB    CAD        RB         UNC
2376          3550  32647777777       RB    SUB        RB
2377                                * STORE 6 WORD BINARY
2378          3551  26137777557 DB6A  OPND  ADD        BSP0  WRD
2379          3552  32177777437       RB    ADD        BUS   DATA
2380          3553  37136777555   SP0       INC        BSP0  WRD
2381          3554  31177777437       RC    ADD        BUS   DATA
2382          3555  37136777555   SP0       INC        BSP0  WRD
2383          3556  30177777437       RD    ADD        BUS   DATA
2384          3557  37136777555   SP0       INC        BSP0  WRD
2385          3560  37177777420   PL        ADD        BUS   DATA
2386          3561  37136777555   SP0       INC        BSP0  WRD
2387          3562  34177777437       DL    ADD        BUS   DATA
2388          3563  37176777555   SP0       INC        BUS   WRD
2389          3564  22177777437       DB    ADD        BUS   DATA
2390                                * CVDB COMPLETION PART - SETS CCE IF 0 (SP2),RESTORE REGS,SDEC
2391          3565  35777417777 DBEN  SP2   ADD                   NZRO      SP2=0 IF RESULT =0
2392          3566  37777777737             ADD              CCE
2393          3567  37762171476             JSB  RSTI              NE2       RESTORE PL,DL,DB,0 IF LONG
2394          3570  00761400020 DPOP  CIR   ROMN             0020 ZERO      SDEC=2
2395          3571  23471777777       SM    ROM        SM    177777           SDEC=3
2396          3572  16471777776       UBUS  ROM        SM    177776
2397          3573  37777757777             ADD                   NEXT
```

```
 ADDRESS    CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

2398
2399                              * 4 WORD RESULT - LONG DIC # 17 OR 18
2400        3574  30302362335     DB4L       RD   JSB  D4B  SP1        UNC       CONV MS WORD 4 DIGITS
2401        3575  16677777777                UBUS ADD       RA
2402        3576  37777777760          PL         ADD
2403        3577  16302362335                UBUS JSB  D4B  SP1        UNC       CONV 2ND WORD
2404        3600  37762363745                     JSB  DBM1             UNC       COMB WORDS 1,2
2405        3601  34302362335                DL   JSB  D4B  SP1        UNC       CONV 3RD WORD
2406        3602  37762363737                     JSB  DBM2             UNC       COMB WORDS 1,2,3
2407        3603  22302362335                DB   JSB  D4B  SP1        UNC       CONV 4TH WORD
2408        3604  37762363731                     JSB  DBM3             UNC       COMB WORDS 1,2,3,4
2409        3605  21302362335                0    JSB  D4B  SP1        UNC       CONV 5TH WORD
2410        3606  37762363723                     JSB  DBM4             UNC       COMPL CONV.
2411        3607  37617777760          PL         ADD       FD                   SHIFT LEFT 1-W
2412        3610  30637777777                RD   ADD       FC
2413        3611  31657777777                RC   ADD       FB
2414        3612  32677777777                RB   ADD       FA
2415                              * COMMON PART OF 4 WORD CASES
***  WARNING ( 2)  ***  RBR CONFLICTS WITH PREFETCH ON INSTR ENTRY
2416        3613  03777437017     DBW4       RBR  ADD            ABS  ODD       FOR SIGN - ABS=1
2417        3614  37766363624                     JMP  DB4A             UNC       POSITIVE SKIP COMPL.
2418        3615  30607507777                RD   SUB       FD        CRRY
2419        3616  31627367777                RC   CAD       FC        UNC
2420        3617  31627507777                RC   SUB       FC        CRRY
2421        3620  32647367777                RB   CAD       FB        UNC
2422        3621  32647507777                RB   SUB       FB        CRRY
2423        3622  33667367777                RA   CAD       FA        UNC
2424        3623  33667777777                RA   SUB       FA
2425                              *STORE 4 WORD RESULT
2426        3624  26137777557     DB4A       OPND ADD       BSP0 WRD
2427        3625  33177777437                RA   ADD       BUS  DATA
2428        3626  37136777555          SP0        INC       BSP0 WRD
2429        3627  32177777437                RB   ADD       BUS  DATA
2430        3630  37136777555          SP0        INC       BSP0 WRD
2431        3631  31177777437                RC   ADD       BUS  DATA
2432        3632  37176777555          SP0        INC       BUS  WRD
2433        3633  30177777437                RD   ADD       BUS  DATA
2434        3634  37766363565                     JMP  DBEN             UNC       COMMON COMPLETE
2435                              * 4 WORD CASE  - SP2<17
2436        3635  37766163650     DB4S            JMP  DB4B             F2        IF RA,RB,RC,FD
2437        3636  37777777760          PL         ADD
2438        3637  16302362335                UBUS JSB  D4B  SP1        UNC       PL,DL,DB,0 CASE
2439        3640  16677777777                UBUS ADD       RA
2440        3641  34302362335                DL   JSB  D4B  SP1        UNC       CONV 2ND WORD
2441        3642  37762363745                     JSB  DBM1             UNC       COMB WORDS 1,2
2442        3643  22302362335                DB   JSB  D4B  SP1        UNC       CONV 3RD WORD
2443        3644  37762363737                     JSB  DBM2             UNC       COMB WORDS 1,2,3
2444        3645  21302362335                0    JSB  D4B  SP1        UNC       CONV 4TH WORD
2445        3646  37762363731                     JSB  DBM3             UNC       COMPLETE CONV
2446        3647  37766363613                     JMP  DBW4             UNC       COMMON 4 WORDS
```

```
2447
2448                                  * 4 WORD RA,RB,RC,RD
2449          3650  33302362335   DB4B      RA    JSB   D4B   SP1         UNC     CONV MS WORD
2450          3651  16677777777             UBUS  ADD         FA
2451          3652  32302362335             RB    JSB   D4B   SP1         UNC     CONV 2ND WORD
2452          3653  37762363745             JSB   DBM1                    UNC     COMB WORDS 1,2
2453          3654  31302362335             RC    JSB   D4B   SP1         UNC     CONV 3RD WORD
2454          3655  37762363737             JSB   DBM2                    UNC     COMB WORDS 1,2,3
2455          3656  30302362335             RD    JSB   D4B   SP1         UNC     CONV 4TH WORD
2456          3657  37762363731             JSB   DBM3                    UNC     COPLETE CONV
2457          3660  37766363613             JMP   DBW4                    UNC     COMMON 4 WORDS
2458                                  * 2 WORDS CASE
2459          3661  32677077777   DBW2      RB    DCAD        FA          NOFL    TEST FOR ILLEGAL DIGIT
2460          3662  37766363770             JMP   TP15                    UNC     NOTE JMP HERE SET SP2_RB
2461          3663  31302362335             RC    JSB   D4B   SP1         UNC     CONV 2ND WORD
2462          3664  37762363745             JSB   DBM1                    UNC     COMB WORDS 1,2
2463          3665  30302362335             RD    JSB   D4B   SP1         UNC     CONV 3RD WORD
2464          3666  37762363737             JSB   DBM2                    UNC     COPLETE CONV
2465          3667  03777437017             RBR   ADD               ABS   ODD     FOR SIGN - ABS=1
2466          3670  37766363674             JMP   *+4                     UNC     SKIP COPL IF POS
2467          3671  31627507777             RC    SUB         PC          CRRY
2468          3672  32647367777             RB    CAD         PB          UNC
2469          3673  32647777777             RP    SUB         PB
2470                                  * STORE 2 WORDS
2471          3674  26137777557             OPND  ADD         PSP0  WRD
2472          3675  32177777437             RB    ADD         BUS   DATA
2473          3676  37176777555       SP0         INC         BUS   WRD
2474          3677  31177777437             RC    ADD         BUS   DATA
2475          3700  37766363565             JMP   DBEN                    UNC     COMMON EXIT
2476                                  * 1 WORD CASE
2477          3701  30302362335   DBW1      RD    JSB   D4B   SP1         UNC     CONV 1 WORD
2478          3702  03777427017             RBR   ADD               ABS   EVEN    FOR SIGN - ABS=1
2479          3703  25527777777             SP3   SUB         SP3             COMPLEMENT
2480          3704  26177777557             OPND  ADD         BUS   WRD       STORE
2481          3705  25177777437             SP3   ADD         BUS   DATA        1 WORD
2482          3706  37766363565             JMP   DBEN                    UNC     COMMON EXIT
```

```
ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

2483
2484                            * SUBROUTINE USED FOR 10K MULTIPLICATION
2485                            * 10K IN Y - MULTIPLICAND
2486                            * DEPENDING ENTRY (IE.DBM1,DBM2,...) 1,2,...REGISTERS ARE
2487                            * MULTIPLIED BY 10K. THE CONTENT OF SP3 IS ADDED TO THE
2488                            * LEAST SIGNIFICANT WORD (REGISTER)
2489   3707  34537417777        DBM6      DL    ADD         SP3        NZRO       6 REGS * 10K
2490   3710  25446363715                   SP3   JMP  DBM5 DB         UNC        DL=0
2491   3711  17772607777                   SBUS  REPN                  20
2492   3712  16774333275             SP0   UBUS  MPAD SP1         INCT CTRM       DL * 10K
2493   3713  17537777777                   SBUS  ADD         SP3                  MS WORD
2494   3714  25457777777                   SP3   ADD         DB                   LS WORD
2495   3715  37537417760        DBM5 PL          ADD         SP3        NZRO       5 REGS * 10K
2496   3716  25706363723                   SP3   JMP  DBM4 PL         UNC        PL=0
2497   3717  17772607777                   SBUS  REPN                  20
2498   3720  16774333275             SP0   UBUS  MPAD SP1         INCT CTRM       PL * 10K
2499   3721  17537777777                   SBUS  ADD         SP3                  MS WORD
2500   3722  25717777777                   SP3   ADD         PL                   LS WORD
2501   3723  30537417777        DBM4      RD    ADD         SP3        NZRO       4 REGS * 10K
2502   3724  25226363731                   SP3   JMP  DBM3 PL         UNC        RD=0
2503   3725  17772607777                   SBUS  REPN                  20
2504   3726  16774333275             SP0   UBUS  MPAD SP1         INCT CTRM       RD * 10K
2505   3727  17537777777                   SBUS  ADD         SP3                  MS WORD
2506   3730  25237777777                   SP3   ADD         PL                   LS WORD
2507   3731  31537417777        DBM3      RC    ADD         SP3        NZRO       3 REGS * 10K
2508   3732  25606363737                   SP3   JMP  DBM2 RD         UNC        RC=0
2509   3733  17772607777                   SBUS  REPN                  20
2510   3734  16774333275             SP0   UBUS  MPAD SP1         INCT CTRM       RC * 10K
2511   3735  17537777777                   SBUS  ADD         SP3                  MS WORD
2512   3736  25617777777                   SP3   ADD         RD                   LS WORD
2513   3737  32537417777        DBM2      RB    ADD         SP3        NZRO       2 REGS * 10K
2514   3740  25626363745                   SP3   JMP  DBM1 RC         UNC        RB=0
2515   3741  17772607777                   SBUS  REPN                  20
2516   3742  16774333275             SP0   UBUS  MPAD SP1         INCT CTRM       RB * 10K
2517   3743  17537777777                   SBUS  ADD         SP3                  MS WORD
2518   3744  25637777777                   SP3   ADD         RC                   LS WORD
2519   3745  33537417777        DBM1      RA    ADD         SP3        NZRO       1 REG * 10K
2520   3746  25657707777                   SP3   ADD         RB         RSB        RET IF RA=0
2521   3747  17772607777                   SBUS  REPN                  20
2522   3750  16774333275             SP0   UBUS  MPAL SP1         INCT CTRM       RA * 10K
2523   3751  17677777777                   SBUS  ADD         RA                   MS WORD
2524   3752  25657707777                   SP3   ADD         RB         RSB        LS WORD
2525                            * DBWC DETERMINES TARGET WORD @LSW FOR CVDB
2526                            * SOURCE DIG # RECEIVED IN RA - WORD @LSW
2527                            * RETURNED IN SP3
2528   3753  33771527773        DBWC      RA    ROM                 7773 POS        -5
2529   3754  25317707777                   SP3   ADD         SP1        RSB        SET 0
2530   3755  33771527756                   RA    ROM                 7766 POS        -10
2531   3756  25316707777                   SP3   INC         SP1        RSB        SET 1
2532   3757  25311600003                   SP3   ROM         SP1  000003
2533   3760  33771527755                   RA    ROM                 7755 POS        -19
2534   3761  37777707777                         ADD                    RSB        SET 3
2535   3762  25311600005                   SP3   ROM         SP1  000005
2536   3763  37777707777                         ADD                    RSB        SET 5
```

```
     ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS                FRI, JUN  4, 1976, 10:14 AM

2537                         * TRAPS FOR CVDB
2538      3764  37531600017  TD17            ROM        SP3  000017      INV DIGIT #
2539      3765  37762361374                  JSB  PSHA            UNC     PUSH 4 TOS REGS
2540      3766  37766363772                  JMP  TDUT            UNC     CVDB TRAP CONT
2541                         * D4B JMPS HERE FOR INVALID DECIMAL
2542                         * FOR CVDB (020601) GOES TO TD15
2543                         * FOR MPYD (020614) GOES TO TA15
2544      3767  00766021456  TCID      CIR  JMP  TA15            EVEN    MPYD
2545      3770  37531600015  TD15            ROM        SP3  000015      INVALID DEC DIG
2546      3771  37762171476                  JSB  RSTL            NE2     RESTORE PL,DL,DB,0 IF LONG
2547      3772  24777777777  TDUT      STA  ADD                           USER TRAP ?
2548      3773  16777522616       UBUS UBUS ADD  SL1       SOV  POS
2549      3774  37571603134                  ROM        PAR  TPP0
                                                                             YES
2550      3775  37766363570                  JMP  DPOP            UNC     DO SDEC
2551                         * FOR CVDB ZERO DIGIT #
2552      3776  37762361374  DBZL            JSB  PSHA            UNC     PUSH 4 TOS REGS
2553      3777  37766363570                  JMP  DPOP            UNC     SDEC & EXIT
2554                         SSTAT
                             #

                ROM COUNT=1964     ERRORS=0          WARNINGS=80
```

```
10CM  2520
9CMP  1361  <=  2433  2516
ADDD  2403  <=  2371
ADDN  2535  <=  2437  2444  2451  2456  2463  2470  2475  2502
ADFN  2531  <=  2515
ADSC  2412
ADSH  2463  <=  2436
ADZL  2537  <=  2035  2036
APOP  2220  <=  1464  2534  2540  2565  2756
BD1W  3260  <=  3155  3251
BD2W  3251  <=  3157  3240
BD3W  3240  <=  3161  3225
BD4W  3225  <=  3163  3207
BD5W  3206  <=  3165  3167
BDEA  3265  <=  3224
BDEC  3263  <=  3237  3250  3257
BDEN  3266  <=  3205
BDFT  3105  <=  3154  3156  3160  3162  3164  3166
BDZL  3306  <=  3120  3121
BND2  1752
BPOP  3302  <=  1650  1651  1777  3307  3322
CAD1  1667  <=  1663
CAD2  1723  <=  1736
CAD3  1725  <=  1711
CAD4  1727  <=  1713
CAD5  1731  <=  1753  1760
CAD6  1743  <=  1720  1730
CAD7  1744  <=  1701  1705  1724  1731
```

```
CAD8  1761  <=  1751
CAD9  1773  <=  1766
CDG*  2143  <=  2267
CKA*  2062  <=  1652  3335
CKAB  2060  <=  2255  2404  2567  3126
CKB*  2041  <=  1660  3340
CKBB  2037  <=  3462
CKD*  2145  <=  2663
CKDG  2144
CKLN  2027  <=  2403  2566
CMP   2541  <=  2507
CPEN  2564  <=  2544  2554  2557
CPOP  3437  <=  3334  3447
CVAD  1641  <=  2362
CVBD  3112  <=  2364
CVDA  3323  <=  2363
CVDB  3450  <=  2365
D0    3062  <=  3236  3247  3256
D1    3055  <=  3202  3204  3220  3222  3234  3245  3254  3260
D2    3050  <=  3200  3216  3232  3243  3252
D3    3043  <=  3176  3214  3230  3241
D4    3036  <=  3174  3212  3226
D4B   2335  <=  2630  2632  2634  2636  3514  3516  3520  3523  3525  3527  3531  3574  3577  3601  3603  3605  3637  3641  3643
                3645  3650  3652  3654  3656  3663  3665  3701
D5    3031  <=  3172  3210
D6    3024  <=  3170
DA1   3361  <=  3356  3401  3404
DA2   3365  <=  3360
```

```
DA3    3400  <=  3374
DA4    3405  <=  3372
DA5    3423  <=  3415
DA6    3426  <=  3411  3421  3423
DA7    3433  <=  3427
DB4A   3624  <=  3614
DB4B   3650  <=  3635
DB4L   3574  <=  3512
DB4S   3635  <=  3510
DB6A   3551  <=  3534
DBEN   3565  <=  3634  3700  3706
DBM1   3745  <=  2633  3517  3600  3642  3653  3664  3740
DBM2   3737  <=  2635  3521  3602  3644  3655  3666  3732
DBM3   3731  <=  2637  3524  3604  3646  3657  3724
DBM4   3723  <=  3526  3606  3716
DBM5   3715  <=  3530  3710
DBM6   3707  <=  3532
DBW1   3701  <=  3504
DBW2   3661  <=  3506
DBW4   3613  <=  3647  3660
DBWC   3753  <=  3463
DBZL   3776  <=  3455
DMP0   2757  <=  2612
DMP1   2617  <=  2770
DMP2   2771  <=  2621
DMP3   2627  <=  2777
DMP4   2672  <=  2725
DMP5   2674  <=  2721
```

```
DMP6  2704  <=  2722
DMP7  2705  <=  2673
DMP8  2723  <=  2710
DMP9  2726  <=  2707
DMPY  1275  <=  2361
DPOP  3570  <=  3775  3777
EAS1  0030  <=  0250
EAS2  0035  <=  0253
EAS4  0057  <=  0054
EAS5  0067  <=  0064
EAS6  0074  <=  0071
EAS7  0113  <=  0103
EAS8  0124  <=  0116
EAS9  0130  <=  0123
EASB  0023
ECMP  0701
ECP5  0740  <=  0213  0745
ECP7  0741  <=  0726  0734  0737
ED10  0472
ED12  0524  <=  0533
ED20  0535  <=  0523
ED22  0561  <=  0566  0641  0643
ED26  0634  <=  0540
ED27  0644  <=  0640
ED30  0570  <=  0560
ED32  0606  <=  0611  0655  0657
ED34  0613  <=  0605
ED36  0651  <=  0573
```

```
ED37  0645  <=  0654  0660
ED40  0614
ED47  0647  <=  0615
ED50  0620  <=  0650
EDIV  0426  <=  0256
EDZ2  0666  <=  0663
EDZR  0661  <=  0434
EFDV  0214  <=  0210
EFV1  0215  <=  0674
EMP2  0314  <=  0311
EMPY  0243
ENEG  0747  <=  0705
ENG2  0750  <=  0761  0765  0766
ENG4  0753  <=  0747
FRAD  1327  <=  2406
FREB  1332  <=  3471
FREE  1331  <=  2106  2263
FRLG  1335  <=  1330  2572  3127
FTA'  1525  <=  1540
FTAG  1526  <=  1532
FTCH  1507  <=  2141  2266  2411  2622  2650  2771  3472
FTDN  1541  <=  1527
L1D   1027  <=  2332  2661  2750  3271
L1D'  1024  <=  2164
L2D   1217  <=  2333  2417
L2D'  1215  <=  2170
L3D   1122  <=  2662  2747  3273
L3D'  1120  <=  2172
```

```
L4D'  2176  <=  2212
MPYD  2566  <=  2374
MSKA  2107  <=  2264  2407  2573  3137
MSKB  2124  <=  2265  2410  2574  3470
NOR2  0152
NOR3  0154  <=  0425  0633
NOR6  0162  <=  0151
NOR7  0167  <=  0161
NORM  0150
NSL'  2232  <=  2140
NSLD  2023  <=  2367
PACK  0171  <=  0160  0166  0236
PSHA  1374  <=  1377  1454  1644  2537  3306  3310  3326  3765  3776
PSHM  0770  <=  0024  0245
R1D   1064  <=  2227  2276  2625  3477
R2D   1246  <=  2230  2302  2415
R3D   1155  <=  2304  2624  2772  3501
R4D   2310
R4D'  2311  <=  2320
REST  2217  <=  2331
RSTA  1465  <=  1460  2564
RSTL  1476  <=  2217  2532  2754  3300  3567  3771
SCA1  1444  <=  1450
SCAN  1421  <=  1410  1416  1442  1447  1451
SCB1  1412  <=  1417
SCNA  1433  <=  2610
SCNB  1401  <=  2232  2607
SLD   2024  <=  2366
```

```
SLDX  2141  <=  2233  2240  2250
SLSR  2227  <=  2157
SLXW  2166  <=  2163
SLXX  2173  <=  2161  2165
SLXY  2206  <=  2200
SLXZ  2213  <=  2175  2231
SPLT  1346  <=  1655  1661  2103  2262  2405  2571  3136  3336  3342  3467
SPT1  1355  <=  1351
SRD   2251  <=  2370
SRLD  2025  <=  2251
SRSL  2332  <=  2271
SRXW  2300  <=  2275
SRXX  2305  <=  2273  2277
SRXY  2321  <=  2307  2334
STMS  1574  <=  1564
STOR  1557  <=  2214  2330  2531  2753  3277
STRS  1606  <=  1614  1620
STSC  1563  <=  1567  1573
STSN  1621  <=  1610
SUBD  2402  <=  2372  2373
TA13  1457  <=  2242  2614  2674  2704  2760
TA15  1456  <=  2146  2150  2152  2154  2443  2450  2455  2462  2467  2474  2501  2506  3767
TA17  1452  <=  2034
TAUT  1461  <=  1455
TB13  3315  <=  3301
TBLN  3310  <=  3117
TBUT  3317  <=  3314
TC15  3443  <=  3370
```

```
TC17  3442  <=  3330
TCID  3767  <=  2336
TD15  3770  <=  3662
TD17  3764  <=  3454
TDUT  3772  <=  3766
TE14  3316  <=  1707  1715  1722  1737  1742
TE17  3313  <=  1647
TPP0  3134
TX13  1453  <=  2533  2755
UAN1  0225  <=  0120
UAN2  0235  <=  0226  0230
UNIM  2400  <=  2360
ZAN1  0224
ZAN2  0231  <=  0224  0022
ZAN3  0020  <=  0313  0323  0464
ZR23  0222  <=  0101  0145
```

<<<<<<<<<RBUS>>>>>>>>>

	0017	1392	MREG	0003	5	PADD	0004	0	PL	0000	36	RA	0013	34	RB	0012	30
RBUS	0005	42	RC	0011	38	RD	0010	29	SP0	0015	121	SP1	0014	33	SR	0001	22
UBUS	0016	125	X	0006	38	XC	0007	0	Z	0002	19						

<<<<<<<<<SBUS>>>>>>>>>

	0037	591	CC	0027	0	CTR	0000	14	CPX1	0004	0	CPX2	0006	0	CTRH	0015	1
CTRL	0014	35	DB	0022	47	DL	0034	45	IOA	0011	0	IOD	0012	0	MOD	0005	0
OPND	0026	131	P	0020	0	PADD	0002	1	PB	0036	19	PCLK	0013	0	Q	0021	26
ODWN	0010	3	RA	0033	88	RB	0032	81	RBR	0003	16	RC	0031	80	RD	0030	74
SBUS	0017	62	SM	0023	63	SP1	0001	106	SP2	0035	81	SP3	0025	131	STA	0024	12
SWCH	0007	0	UBUS	0016	257												

<<<<<<<FUNCTION>>>>>>>

ADD	0037	749	ADDO	0033	0	AND	0007	25	BNDT	0015	19	CAD	0016	58	CADO	0012	0
CAND	0005	1	CRS	0032	31	CTSD	0027	0	CTSS	0034	0	DCAD	0036	36	DVSB	0010	14
INC	0035	127	INCO	0031	0	IOR	0026	48	JMP	0014	242	JSB	0004	193	MPAD	0030	29
PNLR	0020	0	PNLS	0021	0	OASL	0000	22	OASR	0001	30	REPC	0024	4	RFPN	0025	45
ROM	0023	154	ROMI	0022	4	ROMN	0003	45	ROMX	0002	9	SUB	0017	74	SUBO	0013	0
UBNT	0011	0	XOR	0006	5												

<<<<<<<<<STORE>>>>>>>>>

	0037	794	BSP0	0005	61	BSP1	0004	21	BUS	0007	151	CTRH	0017	2	CTRL	0016	28
DB	0022	27	DL	0034	32	IOA	0001	0	IOD	0002	0	MPEC	0003	1	P	0020	0
PB	0036	21	PCLK	0000	0	PL	0011	30	PUSH	0010	4	Q	0021	27	QUP	0013	0
RA	0033	65	RAR	0027	12	RB	0032	74	RC	0031	77	RD	0030	69	SBR	0006	5
SM	0023	10	SP0	0015	49	SP1	0014	110	SP2	0035	104	SP3	0025	142	STA	0024	2
X	0026	34	Z	0012	12												

<<<<<<<<<SKIP>>>>>>>>>

	0037	881	BIT6	0005	3	BITR	0006	2	CRRY	0010	29	CTRM	0033	80	EVEN	0002	19
F1	0014	16	F2	0016	17	F3	0034	19	IUDR	0024	0	JLUI	0031	0	NCRY	0011	29
NEG	0013	19	NEXT	0035	15	NF1	0015	24	NF2	0017	27	NOFL	0007	14	NPRV	0026	7
NSME	0004	3	NZRO	0001	54	ODD	0003	21	POS	0012	29	RSB	0030	52	SR4	0022	0
SRL2	0025	1	SRL3	0027	0	SRN4	0023	0	SRNZ	0021	0	SRZ	0020	1	TEST	0032	0
UNC	0036	361	ZERO	0000	59												

<<<<<<<<<SPECIAL>>>>>>>>>

	0037	1193	CCA	0036	7	CCB	0000	0	CCE	0035	6	CCG	0034	8	CCL	0033	9
CCPX	0001	0	CCRY	0025	2	CCZ	0032	3	CF1	0022	13	CF2	0021	21	CF3	0007	9
CLIB	0016	0	CLO	0031	11	CLSR	0002	11	CTF	0006	35	DCSR	0011	6	FHB	0015	9
HBF	0014	16	INCN	0012	2	INCT	0013	81	INSR	0010	11	LBF	0017	8	POP	0027	2
POPA	0026	2	SCRY	0024	7	SDFG	0005	0	SF1	0023	10	SF2	0020	7	SF3	0003	10
SIFG	0004	0	SOV	0030	4												

<<<<<<<<<MCU>>>>>>>>>

ABS	0000	15	CMD	0002	0	CPL	0001	0	DATA	0021	62	DR	0020	1	DROP	0022	0
NIR	0011	0	OPND	0031	13	PB	0010	0	RND	0024	0	RNP	0014	0	RNS	0034	0
RDA	0007	0	RDD	0027	67	RDND	0023	0	RDNP	0013	0	RDNS	0033	0	RDP	0017	0
RDS	0037	27	RDSA	0005	0	RDSD	0025	0	S	0030	1	WPA	0006	0	WPD	0026	26
WRS	0036	37															

<<<<<<<<<SHIFT>>>>>>>>>

	0007	1251	LLZ	0001	24	LRZ	0000	13	RLZ	0005	11	RRZ	0004	25	SL1	0002	79
SR1	0003	97	SWAB	0006	29												

HP 3000 SERIES II COMPUTER SYSTEM

MICROPROGRAMMING LANGUAGE DESCRIPTION

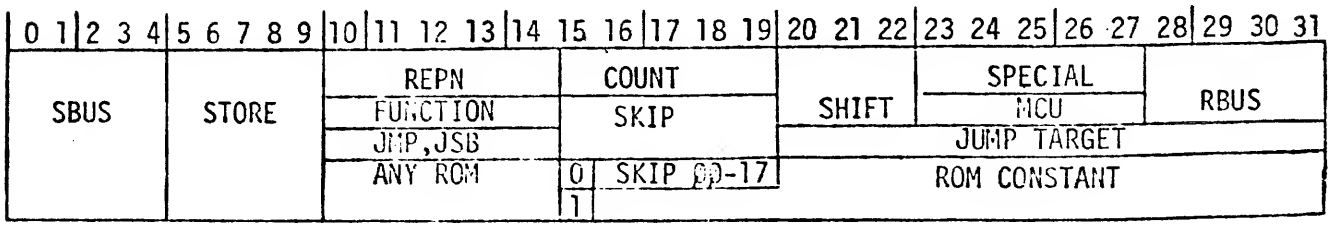
March 1976

8/8/73	RBUS	SBUS	FCN.	SHIFT	STORE	SPEC.	SKIP	MCU	
00	PL	CIR	QASL	LRZ	PCLK	CCB	ZERO T	ABS	00
01	SR	SP1	QASR	LLZ	IOA	CCPX	NZRO T	CRL	01
02	Z	PADD	ROMX	SL1	IOD	CLSR	EVEN	CMD	02
03	MREG	RBR *	ROMN	SR1	MREG	SF3	ODD		03
04	PADD	CPX1	JSB	RRZ	BSP1 *	SIFG	NSME T		04
05	RBUS	MOD	CAND	RLZ	BSPØ *	SDFG	BIT6	ROSA	05
06	X	CPX2	XOR	SWAB	SBR *	CTF	BIT8	WRA	06
07	XC	SWCH	AND	NOP	BUS *	CF3	NOFL	ROA	07
10	RD	QDWN	DVSB		PUSH	INSR	CRRY	PB	10
11	RC	IOA	UBNT		PL	DCSR	NCRY	NIR	11
12	RB	IOD	CADO T		Z	INCN	POS		12
13	RA	PCLK	SUBO T		QUP	INCT	NEG	RONP	13
14	SP1	CTRL	JMP		SP1	HBF	F1	RNP	14
15	SPØ	CTRH	BNDT		SPØ	FHB	NF1		15
16	UBUS	UBUS	CAD		CTRL	CLIB	F2		16
17	NOP	SBUS	SUB		CTRH	LBF	NF2	ROP	17
20		P	PNLR+		P	SF2	SRZ	DB	20
21		Q	PNLS+		Q	CF2	SRNZ	DATA	21
22		DB	ROMI		DB	CF1	SR4	DPOP	22
23		SM	ROM +		SM	SF1	SRN4	ROND	23
24		STA	REPC+		STA	SCRY	INDR	RND	24
25		SP3	REPN+		SP3	CCRY	SRL2	ROSD	25
26		OPND	IOR		X	POPA T	NPRV	WRD	26
27		CC	CTSD+		RAR	POP	SRL3	ROD	27
30		RD	MPAD+		RD	SCV	RSB	S	30
31		RC	INCO+T		RC	CLO	JLUI	OPND	31
32		RB	CRS +		RB	CCZ T	TEST		33
33		RA	ADDO+T		RA	CCL	CTRM	RONS	33
34		DL	CTSS+		DL	CCG	F3	RNS	34
35		SP2	INC +		SP2	CCE	NEXT		35
36		PB	DCAD+		PB	CCA T	UNC	WRS	36
37		NOP	ADD +		NOP	NOP	NOP	ROS	37

* These options inhibit execution of the "SPEC" field options and enable the "MCU" field options in their place.

\+ These functions cause an "ADD".

T Test is made on the T-bus.



A BRIEF EXPLANATION OF THE MICROINSTRUCTION FIELDS

There are nine fields and a comment space in the microinstruction, which are coded as follows:

coding sheet

1---4	6---9	11---14	16---19	21---24	26---29	31---34	36---39	46------72
LABEL	R-BUS	S-BUS	FUNC.	SHIFT	STORE	SPEC	SKIP	COMMENTS

The LABEL field may contain any characters in columns 1-4 provided that the first character is not the blank character, an asterisk (*), an ampersand (&), or a percent sign (%). These are interpreted by the assembler as follows:

blank - No label.

& - If followed by a 4 digit octal number (e.g. 0271) in col's. 2-5, the assembler will load this number in its address counter and continue the assembly from there. If not followed by a number, the assembler address counter is set to the beginning of the next 256 word sector, and the assembly continues from there.

* - Indicates a comment card which will appear on the listing, but will not affect the assembly.

% - May be used to insert a label in the symbol table. This is useful when assembling code segments that refer to labels in other segments not being assembled. The format is %bxxxxbbyyyy where:

b = blank
xxxx = 4 character (max.) label (trailing characters on labels of less than 4 characters are blanks).
yyyy = 4 character octal address of the label in the sector referenced. Must contain 4 characters and no blanks (e.g. Ø12Ø).

NOTE: To "NOP" a field, it must be left blank. Note that there is no NOP in the function field; hence this field must always be coded with something.

The R-BUS field in columns 6-9 points to the register to be placed in the R-BUS register.

The S-BUS field in columns 11-14 points to the register to be placed in the S-BUS register.

The FUNCTION field in columns 16-19 gives the function that the arithmetic logic unit (ALU) is to perform on the two operands contained in the R-BUS and S-BUS registers or a special function.

The SHIFT field in columns 21-24 denotes how the information resulting from the ALU and placed on the T-BUS should be shifted and placed on the U-BUS.

The STORE field in columns 26-29 points to a register in which the contents of the U-BUS are to be stored.

The SPECIAL field in columns 31-34 has many varied uses which are best explained in that section of this document.

The SKIP field in column 36-39 denotes conditions of the CPU on which logical decisions in the microprogram can be made.

The COMMENT field may contain any explanatory comments.

JMP, JSB Targets are 4-character alphanumeric labels coded in place of the SHIFT field (col's 21-24 of the coding sheet). The μ-assembler matches this label with a binary address (12-bit) from the symbol table and inserts this address in bits 20-31 of the ROM word containing the JMP or JSB. This voids the SHIFT, SPEC, & RBUS fields (must be coded with LABEL, NOP NOP, respectively).

ROM functions (ROM, ROMI, ROMX, ROMN) cause a 16-bit constant to be placed in the RBUS register, which is then operated on by the ALU in conjunction with the SBUS register as explained in the section on "function field". The constant is coded as a 4- or 6- (octal) digit in place of the SPEC field (col's. 23-26 or 23-28 of the coding sheet). If a 4-digit number is detected by the μ-assembler, a 12-bit number corresponding to the octal number is placed in the ROM word in positions 20-31. In addition ROM(15) is set to Ø. This enables skips ØØ-%17 (i.e. ZERO-NF2) of the skip field (note that a skip must be coded- NOP is not possible). When the μ-instruction is executed in the machine, the constant is placed right-adjusted into the RBUS register, with the hardware extending the sign of the 12-bit number (bit 4) into bits Ø-3 of the register (e.g. %7774 becomes 177774). The SHIFT, SPEC, & RBUS fields are voided (must be coded NOP, constant, NOP, respectively). If a 6-digit number is detected by the μ-assembler, a 16-bit number corresponding to this octal number is placed in the ROM word in positions 16-31. ROM(15) is also set to a 1. In this case the SKIP, SHIFT, SPEC & RBUS fields are voided (must be coded NOP, NOP, constant, NOP respectively - note also that the octal constant spills over into the skip field when coding). In both cases, the μ-code listing will show the 6-digit (octal) number.

TOP OF THE STACK

The stack has a topmost element which is LOGICALLY the quantity A. Similarly, there is a LOGICAL quantity B, C, and D corresponding to the second, third, and fourth word of the stack, respectively. The LOGICAL quantities A, B, C, and D may be either in registers or in memory. This is determined by the SR register. If the SR register is 0 then none of the logical quantities A, B, C, or D are in registers but rather they are located in memory locations (SM), (SM-1), (SM-2), and (SM-3), respectively.

At all times however, there are four registers RA, RB, RC, and RD, which are named by a hardware naming device. In the microprogram the micro-options RA, RB, RC, and RD refer to the hardware named registers and NOT TO THE LOGICAL QUANTITIES A, B, C, and D. There is a correspondence however. For any of the LOGICAL quantities A, B, C, and D, the state of SR indicates where it is located by the following table:

SR	A	B	C	D
0	(SM)	(SM-1)	(SM-2)	(SM-3)
1	RA	(SM)	(SM-1)	(SM-2)
2	RA	RB	(SM)	(SM-1)
3	RA	RB	RC	(SM)
4	RA	RB	RC	RD

Note then that if SR=1, B is in (SM) and if the micro-op RB is used, the contents of the register named RB will be affected, NOT THE LOGICAL quantity B (i.e. for this case, RB, RC, and RD could be used as scratch pads without affecting B, C, or D).

The micro-store field instruction PUSH does three things:

1. Stores the output of the shifter into the register RD.
2. Increments the SR register.
3. Renames the registers so that

   N(RA) := RB, N(RB) := RC, N(RC) := RD, N(RD) := RA where N(RA) is read 'the register named RA' (i.e., N(RA) := RB is read 'the register named RA becomes RB').

   This "pushes" the contents of the U-BUS onto the top of the stack (TOS).

Similarly the micro-spec field instruction POP does two things:

1. Decrements the SR register.
2. Renames the registers so that
   N(RA) := RD, N(RB) := RA, N(RC) := RB, N(RD) := RC.
   This "pops" the top element from the stack.

The micro-functions QUP, QDWN, MREG read and stores, are fully explained in the field descriptions and should be used very seldomly since the stack will be preadjusted in most cases.

R-BUS Field

(blank) Zero is placed in the R-BUS register.

MREG SP1(14:15) are added to the contents of the namer register to get a temporary name. This is used to reference a memory element that happens to lie in the TOS. Register SP1(14:15) contains S-E (where S = SR + SM and E = Effective Address). A TOS register (RA, RB, RC, or RD) used as a store option on the line immediately preceding this instruction will assume this temporary name.

PADD The pre-adder contents are placed in the R-BUS register.

PL The Program Limit register, PL, is placed in the R-BUS register.

RA The register named RA by the hardware namer is placed in the R-BUS register.

RB The register named RB by the hardware namer is placed in the R-BUS register.

RBUS The R-BUS register is unchanged.

RC The register named RC by the hardware namer is placed in the R-BUS register.

RD The register named RD by the hardware namer is placed in the R-BUS register.

SPØ Scratch Pad Ø, SPØ, is placed in the R-BUS register.

SP1 Scratch Pad 1, SP1, is placed in the R-BUS register.

SR The Stack Register counter, SR, is placed in the R-BUS register (13:15), preceded by 13 leading zeros.

UBUS The output of the shifter (i.e., the U-BUS), is placed in the R-BUS register.

X — The index register, X, is placed in the R-BUS register.

XC — If the index bit of the current instruction is zero, then 0 is placed in the R-BUS register, otherwise the index register is placed in the R-BUS register. The index bit, for indexable instructions, = CIR(4).

Z — The stack limit register is placed in the R-BUS register.

S-BUS Field	
(blank)	Zero is placed in the S-BUS register.
*CC	SBUS(8:9) := STATUS(6:7) and if STATUS(6:7) = 00 then SBUS(7) := 1 else SBUS(7) := 0 All other bits of SBUS are zeroed. Note: SBUS = S-BUS register.
CIR	The contents of current instruction register is placed in the S-BUS register.
**CPX1	RUN Mode Interrupt Status register is placed in the S-BUS register. Also clears the I/O Timer FF if no SIO transfer is in process.
**CPX2	HALT Mode Interrupt Status register is placed in the S-BUS register.
*CTRH	S-BUS REG(4:9) := CNTR(0:5). This will be used mostly in floating point exponent manipulations. Note: CNTR is a 6-bit binary counter.
*CTRL	S-BUS REG(10:15) := CNTR(0:5)
DB	The data base register, DB, is placed in the S-BUS register.
DL	The data limit register, DL, is placed in the S-BUS register.
*IOA	The I/O address register is placed in the S-BUS register bits 8:15. (Reads Interrupting Device NO.)
IOD	The I/O data register is placed in the S-BUS register. (Reads Direct Data Buffer.)

*Unless otherwise noted, remaining bits are zero.

**See explanation of interrupts.

MOD A constant is brought to the S-BUS register in the following way:

Left byte: contains transmitted MOP (command) and sender's module number as shown:

```
SBUS(Ø,1,4) := Ø
SBUS(2,3)   := MOP
SBUS(5:7)   := Interrupting module no.
```

Valid only after module interrupt is received, and until the MOD INT FF is cleared.

Right byte: contains encoded CPU No. information as follows

```
SBUS(8:15) := %ØØ4 if CPU #1
SBUS(8:15) := %Ø1Ø if CPU #2
```

OPND The operand register is placed in the S-BUS register.

P The Program counter, P, is placed in the S-BUS register.

PADD The pre-adder output is placed in the S-BUS register.

PB The Program Base register, PB, is placed in the S-BUS register.

PCLK The Process Clock PCLK, is Placed in the S-BUS register.

Q The Stack Marker Pointer register, Q, is placed in the S-BUS register.

QDWN It takes the lowest valid TOS register and puts it in the S-BUS register in the following way: the TOS registers are renamed by NAMER + SR. RD is then dispatched to the S-BUS. The TOS registers are returned to their former names on the following cycle. A TOS register used in the STORE field of the previously executed instruction will assume a temporary name. A DCSR Special Option is needed to complete the operation.

RA The register named RA by the hardware namer is placed in the S-BUS register.

RB The register named RB by the hardware namer is placed in the S-BUS register.

RBR Read bank register onto SBUS(14:15). SBUS(Ø:13) := Ø. The bank register to be read is specified in the "MCU" field. Execution of the "SPEC" field is inhibited.

RC The register named RC by the hardware namer is placed in the S-BUS register.

RD	The register named RD by the hardware namer is placed in the S-BUS register.
SBUS	The S-BUS register is unchanged.
SM	The memory Top of Stack pointer register, SM, is placed in the S-BUS register.
SP1	Scratch Pad register 1, SP1, is placed in the S-BUS register.
SP2	Scratch Pad register 2, SP2, is placed in the S-BUS register.
SP3	Scratch Pad register 3, SP3, is placed in the S-BUS register.
STA	The Status register, STA, is placed in the S-BUS register.
SWCH	The switch register contents are placed in the S-BUS register.
UBUS	The output of the shifter (i.e., the U-BUS) is placed in the S-BUS register.

Function Field

ADD — The contents of the R-BUS and the S-BUS registers are added and the result is placed on the T-BUS.
Note: The T-BUS is the ALU output (or shifter input).

ADDO — The contents of the R-BUS and the S-BUS are added and the result is placed on the T-BUS. The overflow and carry bits in the STATUS word are set or cleared depending on the state of the ALU output. CCA is set from the T-BUS.

AND — The logical AND of the R-BUS and the S-BUS is placed on the T-BUS.

QASL — Causes a 4 register arithmetic shift left of the U-BUS, SP3, SP1 and the R-BUS register containing the most, next most, next least, and least significant word, respectively. SL1 is required in the shift field and the direction of the shift is left. The sign bit is preserved.

```
T-BUS:= SREG;
UBUS(Ø):= TBUS(Ø);
UBUS(1:14):= TBUS(2:15);
UBUS(15);= SP3(Ø);
SP3(Ø:14):= SP3(1:15);
SP3(15):= SP1(Ø);
SP1(Ø:14):= SP1(1:15);
SP1(15):= RREG(Ø);
RREG(Ø:14):= RREG(1:15);
RREG(15):= Ø;
```

QASR — Causes a 4 register arithmetic shift right of the U-BUS, SP3, SP1, and the S-BUS register containing the most, next most, next least, and least significant words respectively. SR1 is required in the shift field and the direction of the shift is right. The sign bit is propagated.

```
TBUS:= RREG;
UBUS(Ø:1):= TBUS(Ø);
UBUS(2:15):= TUBS(1:14);
SP3(Ø):= TBUS(15);
SP3(1:15):= SP3(Ø:14);
SP1(Ø):= SP3(15);
SP1(1:15):= SP1(Ø:14);
SREG(Ø):= SP1(15);
SREG(1:15);= SREG(Ø:14):
```

BNDT The function executes a hardware bounds test of an address. If the shift field contains LRZ, RRZ, RLZ, or LLZ, then

TBUS := RBUS-SBUS -1 (and the shift is executed)

else

TBUS := RBUS-SBUS.

If the ALU Carry Out =1, the next μ-instruction is fetched. If the carry =Ø and the machine is in USER mode, a hardware μ-jump is made to ROM addr. 3. BNDT takes precedence over the skip field if the test fails. The above allows bounds tests to be made for (RBUS) > (SBUS) (1st case) or (RBUS) $\geq$ (SBUS) (2nd case).

CAD The 1's complement of the S-BUS is added to the R-BUS and the result placed on the T-BUS.

CADO Same as CAD with addition that the carry and overflow bits in the STATUS word are set or cleared depending on the state of the ALU output. CCA is set from the T-BUS.

CAND T-BUS := R-BUS AND ($\overline{\text{S-BUS}}$).

CRS The T-BUS is circular shifted right (SR1) or left (SL1) one bit and put on the U-BUS. U(0) := T(15) if SR1, or U(15) := T(0) if SL1. Implied T-BUS := R-BUS + S-BUS.

CTSD This function performs a double register shift of the T-BUS and a scratch pad register. A left shift, indicated by an SL1 in the shift field, expects the least significant word in SP1. A right shift (SR1) expects the least significant word in SP3. The type of shift is determined from the contents of the CIR as follows. T-BUS := R-BUS + S-BUS implied.

CTSD (Cont.)

CIR(7) = 1 Circular shift
CIR(7:8) = 0,1 Logical shift
CIR(7:8) = 0,0 Arithmetic shift

Note: Both SP1 and SP3 get shifted on CTSD. Hence one or the other, depending on the direction of the shift, will contain garbage at the end.

CTSS The T-BUS is shifted in a manner determined by CIR(7:8) as follows. Implied T-BUS := R-BUS + S-BUS.

CIR(7) = 1 Circular shift
CIR(7:8) = 0,1 Logical shift
CIR(7:8) = 0,0 Arithmetic shift

Note: The direction is determined by shift field.

DCAD Adds two 4-digit decimal numbers together and places the result on the U-BUS. For valid results each digit must be in the range $\emptyset \leq n \leq 9$. A carry digit (F3) is added to the least significant digit during the add, and a decimal carry out is saved (in F3) at the end of the add. This allows multiple-register adds. F3 must be cleared prior to the first add in order to obtain valid results.

Specifically. The function DCAD adds the contents of the R- and S-BUS registers and puts the result into a decimal correction adder. the shifter is turned off (inhibiting the ALU output from the U-BUS), and the decimal corr. adder output is placed onto the U-BUS. The decimal carry FF (F3) logic is enabled.

Shift field and spec. field "FHB" are ignored. The T-BUS (ALU output) reflects the result of the uncorrected binary addition.

If an invalid digit is detected in either the R- or S-BUS registers during the add cycle, the "set overflow" line is asserted to provide a skip test indication (the state of the OVFLO FF is not affected). Normal code sequence is, then, (registers are arbitrary)

RA	RB	DCAD	—	SP1	—	NOFL
—	—	JMP	TRAP	—	—	UNC

This function performs the subtract, shift, and test necessary to implement a divide algorithm. To start, F2 = 0, the divisor is in the S-Reg., and the double word dividend is in the R-Reg. (MSW) and SP1. SL1 must be in the shift field. One bit quotient comes in SP1(15).

DVSB ALGOL DEFINITION:

```
    TBUS:= RBUS-SBUS;
    UBUS(Ø:14):= TBUS(1:15);    BY SL1 in shift field
    If ALU carry or F2=1 then
        BEGIN
            RREG(Ø:14) := UBUS(Ø:14);
            RREG(15) := SP1(Ø);
            SP1(Ø:14) := SP1(1:15);
            SP1(15) := 1;
            F2 := TBUS(Ø);
        END
    else
        BEGIN
            RREG(Ø:14) := RREG(1:15);
            RREG(15) := SP1(Ø);
            SP1(Ø:14) := SP1(1:15);
            SP1(15) := Ø;
            F2 := RREG(Ø);
        end;
```

INC T-BUS := R-BUS + S-BUS + 1

INCO — Same as INC with the addition that the carry and overflow bits in the STATUS word are set or cleared depending on the state of the ALU output. CCA is set from the T-BUS.

IOR — The R-BUS and S-BUS are logically ORed together and the result placed on the T-BUS.

JMP — This function performs a micro-jump to the ROM address specified in bits 20 to 31 if the condition contained in the skip field is met. If the skip condition is not met, the next ROM instruction in sequence is fetched. Also, implied U-BUS := T-BUS := S-BUS.

JSB — This function causes a subroutine jump, and is executed like the JMP function except that the RAR register is stored into the SAVE reg. This is the return address for the subroutine. A JSB FF is also set (see also RSB in the skip field).

MPAD — This function performs the shift, test, and add functions necessary to implement a multiply algorithm. To start, the multiplier is in SP3, the multiplicand is in the R-BUS register, and the S-BUS register = 0. An SR1 is required in the shift field. One bit result comes in SP3(0).

```
T-BUS := R-REG + S-REG; U-BUS(1:15) := T-BUS(0:14)
U-BUS(0) := ALU carry; if SP3(15) = 1, then S-Reg :=
U-BUS, SP3(1:15) := SP3(0:14); SP3(0) := T(15); else
S-REG(1:15) :=  S-REG(0:14),SP3(1:15) := SP3(0:14),
SP3(0) := S-REG(15).
```

PNLR — A maintenance panel function. The appropriate register selected from the maint. panel is brought to the T-BUS through ALU. (R- and S-field are ignored). Appropriate bank reg. is gated to the bank lines.

PNLS — A maintenance panel function. The U-BUS is stored in the appropriate register selected from the maintenance panel. If the register selected is PB, DB, Z, or Mem. Addr. (SPØ), The value in the bank switches is stored into the appropriate bank register.

The following two functions are repeat commands and operate in the following manner. The microinstruction following the repeat command is executed over and over until the skip field condition of the repeated instruction is met. The instruction is then terminated and normal microprocessing proceeds. The skip field of the REPN instruction may not be used, except as shown below. The two repeat functions differ only in what they do during their execution, not in the operation of the repeated instruction. A repeated line of μ-code will execute at least once, even if its SKIP condition is immediately met.

REPC — Normal repeat function that has implied T-BUS := R-BUS + S-BUS.

REPN — Send skip field contents to CNTR, CNTR(0) = 1, and implied T-BUS := R-BUS + S-BUS. (The μ-assembler puts -(skip field) into the counter). Note: Skip field tests are inhibited.

See explanation on Page 3A for the following 4 functions.

ROM — Bits 20-31 or 16-31 of this instruction are placed in the R-BUS reg. (If the former, bits 0:3 of this reg. are set to bit 4 (sign extension)). Implied TBUS := RBUS + SBUS.

ROMI — Same as ROM except implied T-BUS := inclusive - OR of R and S BUS.

ROMN — This function is like ROM except implied T-BUS := R-BUS AND S-BUS.

ROMX — This function is like ROM except implied T-BUS := R-BUS XOR S-BUS.

SUB — T-BUS := R-BUS - S-BUS.

SUBO — Like SUB, except carry and overflow bits in the STATUS word are set or cleared depending on the state of the ALU output. CCA is set from the T-BUS.

UBNT — Unconditional bounds test. Same as BNDT except no test for USER mode is made (i.e. if the test fails, the jump to ROM addr. 3 is made regardless of machine mode).

XOR — T-BUS := R-BUS EXCLUSIVE OR S-BUS.

Shift Field

Note: Left byte = bits Ø:7
Right byte = bits 8:15

(blank) No shift, U-BUS := T-BUS.

LLZ "Left to left and zero" places the left byte of the T-BUS in the left byte of the U-BUS and places zeros in the right byte of the U-BUS.

LRZ "Left to right and zero" places the left byte of the T-BUS in the right byte of the U-BUS and places zeros in the left byte of the U-BUS.

RLZ "Right to left and zero" places the right byte of the T-BUS in the left byte of the U-BUS and places zeros on the right byte of the U-BUS.

SWAB "Swap Bytes" places the right byte of the T-BUS in the left byte of the U-BUS and the left byte of the T-BUS in the right byte of the U-BUS.

RRZ "Right to right and zero" places the right byte of the T-BUS in the right byte of the U-BUS and places zeros in the left byte of the U-BUS.

SL1 "Shift left one" shifts the T-BUS one bit left onto the U-BUS. When used with TASL, CTSS, CRS, CTSD and DVSB in the function field, refer to those descriptions to determine the action taken. This option may be used alone to perform a single logical shift where a zero is brought into U-BUS(15) and bit Ø of the T-BUS is lost.

SR1 "Shift right one" shifts the T-BUS one right onto the U-BUS. When used with TASR, CTSS, CRS, CTSD and MPAD in the function field, refer to those descriptions to determine the action taken. This option may be used alone to perform a single logical right shift where a zero is brought into U-BUS(Ø) and bit 15 of the T-BUS is lost.

Store Field

(blank) No store.

BSPØ Stores U-BUS into A-COR or D-COR, depending on the MCU field option selected, and into SPØ. It disables the special field and enables the MCU options, one of which must be used.

BSP1 Same as BSPØ except SP1 is used.

BUS Same as BSPØ, except none of the scratch-pad registers are used.

CTRH Counter high stores U-BUS(4:9) in the counter.

CTRL Counter low stores U-BUS(10:15) in the counter.

DB Stores the U-BUS in the Data Base Register, DB.

DL Stores the U-BUS in the Data Limit register, DL.

IOA Sends the command on UBUS(5:7) to the device whose address is on UBUS(8:15) UBUS(Ø)=1 is used to generate the "service out" signal to the device. UBUS(8) is treated by the hardware as a "don't care" (device addresses are limited to 7 bits, contained in UBUS (9:15)).

IOD Stores the UBUS into the I/O Data register.

MREG The contents of Namer is added to two bits (SP1(14:15)) to obtain temporary name. This is used to reference a memory element that happens to lie in the TOS registers. SP1(14:15) contains E-SM. TOS registers used in the R and S field in the line following this instruction will assume the temporary name.

P Stores the U-BUS into the program counter, P.

PB Stores the U-BUS into the Program Base register, PB.

PCLK Stores the U-BUS into the Process Clock register, PCLK.

PL	Stores the U-BUS into the Program Limit register, PL.
PUSH	Stores the U-BUS into the RD register, increments the SR register by one and at the end of the microinstruction cycle renames the TOS registers such that: N(RA) := RB, N(RB) := RC, N(RC) := RD, N(RD) := RA.
Q	Stores the U-BUS in the Stack Marker Pointer, Q.
QUP	The TOS registers are renamed by NAMER + SR. Temporarily named RA := U-BUS. The TOS register names are returned to NAMER - however incrementing of SR is not implicit: INSR (inc. SR) must appear in the special field in order to increment SR. TOS registers used in the R and S fields following this instruction will assume the temporary name.
RA	Stores the U-BUS in the register named RA.
RAR	Gates U-BUS(Ø:15) onto VBUS(Ø:15). This takes 3 cycles. (See also Appendix B #26). Skip field is ignored.
RB	Stores the U-BUS in the register named RB.
RC	Stores the U-BUS in the register named RC.
RD	Stores the U-BUS in the register named RD.
SBR	Stores U-BUS(14:15) into the bank register specified in the "MCU" field. Execution of the "SPEC" field is inhibited.
SM	Stores the U-BUS into the memory stack pointer, SM.
SPØ	Stores the U-BUS into scratch pad register Ø, SPØ.
SP1	Stores the U-BUS into scratch pad register 1, SP1.
SP2	Stores the U-BUS into scratch pad register 2, SP2.

SP3	Stores the U-BUS into scratch pad register 3, SP3.
STA	Stores the U-BUS into the Status Register.
X	Stores the U-BUS into the index register, X.
Z	Stores the U-BUS into the stack limit pointer, Z.

Special Field

Note: If the S-BUS field contains "RBR", or the STORE field contains "BUS", "BSPØ", "BSP1", or "SBR", then special field is disabled and MCU field is enabled.

(blank) No special option.

CCA Sets the condition code bits in the status word to
CCL if T-BUS < Ø.
CCE if T-BUS = Ø.
CCG if T-BUS > Ø.

CCE Sets the condition code bits in the status word to CCE.
STA(6:7) := 1,Ø

CCG Sets the condition code bits in the status word to CCG.
STA(6:7) := Ø,Ø

CCL Sets condition code bits in status word to CCL.
STA(6:7) := Ø,1

CCPX Clears the interrupt status register bits as specified by the true bits on the U-BUS. (See explanation of interrupts.)

CCRY Clear the carry bit in the status word.

CCZ Sets condition code bits in status word to CCE if T-BUS = Ø and CCG if T-BUS not equal Ø.

CF1 At the end of the cycle, CF1 clears Flag 1.

CF2 At the end of the cycle, CF2 clears Flag 2.

CF3 At the end of the cycle, CF3 clears Flag 3.

CLIB At the end of the cycle, CLIB sets a FF which masks the indirect line until a NEXT or JLUI option in the SKIP field is encountered. This FF may also be cleared by a UBUS(8).CCPX operation (NIR→CIR).

CLO — At the end of the cycle, CLO clears the overflow bit in the status word.

CLSR — Sets the SR register to zero during the cycle. Note that this is an asynchronous reset. No other SR operation during the cycle is allowed.

CTF — Stores the ALU carry in Flag 1 at the end of the cycle.

DCSR — Decrements the SR counter by 1.

INCN — Increments the Namer. N(RA) := RD, N(RB) := RA, N(RC) := RB, N(RD) := RC (Can be read "the register named RA becomes the register named RD, etc.")

FHB — Flag 1 to high bit. U-BUS(0) := FLAG1.

HBF — FLAG1 := U-BUS(0).

INCT — Increments the counter by 1 (modulo 64).

INSR — Increment SR by 1.

LBF — Low bit to flag 2. F2 := U-BUS(15).

POP — This option decrements the SR by 1 and then renames the TOS registers (increments namer) such that:
N(RA) := RD, N(RB) := RA, N(RC) := RB, N(RD) := RC.

POPA — Exactly like POP, except CCA is set on the contents of the T-BUS.

SCRY — Set the carry bit in the status word.

SDFG — Sets the dispatcher flag CPX1(12) := 1.

SF1 — Sets flag 1 at the end of the cycle.

SF2 — Sets flag 2 at the end of the cycle.

SF3 — Sets flag 3 at the end of the cycle.

SIFG Sets the interrupt flag CPX1(11) := 1.

SOV Sets the overflow bit in the status word at the end of the cycle.

CCB Sets CCB on contents of UBUS(8:15):

CCL = Special

CCE = Alphabetic

CCG = Numeric

NOTE: CCL = STA (6:7) = Ø1

CCE = STA (6:7) = 1Ø

CCG = STA (6:7) = ØØ

Skip Field

The skip field does one of two things:

1. Sets the condition met flag or
2. Initiates a hardware micro-jump. A hardware micro-jump needs no jump target in the micro-instruction.

The condition met flag after a REPN or REPC function option indicates the condition on which to terminate the repeated micro-instruction. Otherwise it indicates that the next micro-instruction is to be skipped. (See also the explanation of ROM constants on Page 3a).

(blank) No skip option

*BIT6 Condition met if bit 6 of the U-BUS is a 1.

*BIT8 Condition met if bit 8 of U-BUS is a 1.

*CRRY Condition met if the carry out of the ALU is a one. (Note: This is <u>not</u> the carry bit in the status word.)

CTRM Condition met if the counter contains all ones. (Note: When INCT CTRM options occur the counter is tested before it is incremented.)

*EVEN Condition met if U-BUS(15) = 0.

F1 Condition met if at the beginning of the cycle, flag 1 is set.

F2 Condition met if at the beginning of the cycle, flag 2 is set.

F3 Condition met if at the beginning of the cycle flag 3 is set.

INDR Condition met if the indirect bit of the current instruction register is set, where

$$INDR = (CIR(4) \cdot \overline{MEM\ REF} + CIR(5) \cdot MEM\ REF) \cdot \overline{CLIBFF}$$

JLUI — Conditional hardware microjump to the address that the lookup table is displaying if the indirect line from the CIR is not = 1. CLIB must have been previously used to guarantee a jump on all instructions. "JLUI" resets the CLIBFF at the end of the cycle.

*NCRY — Condition met if the carryout of the ALU is zero. (Note this is <u>not</u> the carry bit in the status word.)

*NEG — Condition met if U-BUS(0) = 1.

NEXT — Terminates current instruction and initiates the sequence necessary to begin execution of the next instruction. If stackop A has just been executed and stackop B is not a NOP, then the hardware executes stackop B. Otherwise the action shown in the timing figure below takes place (a, b, c, d, e, f are equal length CPU clock cycles):

a	b	c	d	e	f
... Mem. Sel. cycle Data→NIR	NIR→LUT	...	NEXT BUSL, RWP Issue LOREQ LUT→VBUS→ ROM→RANK1 NIR→CIR	NOP2 P+1→P Select cycle. RANK1→RANK2 (if mem. ref., force PADD, BASE to R, S-BUS Reg's.	execute 1st line of μ- code of new instr.

Time periods a, b, c (if present), and d occur in the currently executing instruction. "a" and "b" must occur before "d" for maximum execution speed- otherwise a CPU freeze will occur at "d". "a" and "b" result from the "next instruction prefetch" of the current instruction. "c" may or may not be present depending on the length of the current instruction. "d" is the last line of the current instruction. It initiates a "next instruction prefetch", transfers (NIR) to CIR, and applied the address on the VBUS (normally using the LUT output) to the ROM input. The ROM word at this address is stored in RANK1. In addition, the NOP2 FF is set. "e" is used to increment the P-reg., transfer RANK1 to RANK2, and if the new instruction is a memory-reference type, load the R- and S-BUS reg's. with the Pre-adder output and the proper base register. This is also the "select" cycle for the "next instr. prefetch" if there is no MCU conflict. During "f", the first line of the new instruction is executed.

NEXT (Cont.)

The above is the normal sequence of operation of "NEXT". This sequence is modified in the event an interrupt is pending or the μ-code line is "...DATA NEXT".

"NEXT" also clears F1, F2, F3, CNTR, Subroutine Flag FF, and the ABS-BANK reg.

NF1 Condition met if at the beginning of the cycle, flag 1 is cleared.

NF2 Condition met if at the beginning of the cycle, flag 2 is cleared.

NPRV Condition met if at the beginning of the cycle the privileged mode bit is not set.

*NSME Condition met if all the bits of the T-BUS are not the same.

*NZRO Condition met if T-BUS is non-zero.

*ODD Condition met if U-BUS(15) = 1.

*NOFL	Condition met if overflow out of the ALU does not occur. (Note this is not the overflow bit in the status word.) See also DCAD in function field.
*POS	Condition met if U-BUS(0) = 0.
RSB	Hardware micro-jump to the address held in the SAVE register. The SAVE register contents is transferred to the RAR incrementer and the VBUS. If a JSB has not been executed prior to this option, it is treated as a NOP
SR4	Condition met if the SR register is 4.
SRL2	Condition met if the SR register is less than 2.
SRL3	Condition met if the SR register is less than 3.
SRN4	Condition met if the SR register is not 4.
SRNZ	Condition met if the SR register is non-zero.
SRZ	Condition met if the SR register is zero.
TEST	Condition met if any interrupt is pending.
UNC	Condition met unconditionally.
*ZERO	Condition met if the T-BUS is zero.

*These tests are defined to be the data-dependent tests. All other conditions are known at the beginning of the cycle.

MCU Field

This field is executed in place of the "SPEC" field when the "S-BUS" field contains "RBR", or the "STORE" field contains "BUS", "BSPØ", "BSP1", or "SBR".

ABS — Specifies ABSOLUTE bank register. May ve read onto SBUS(14:15) with "RBR" or stored into from UBUS(14:15) with "SBR". This bank register is normally used with instructions requiring absolute addresses.

CMD — Enables the bus option (BUS, BSPØ, BSP1) in the "STORE" field to store the U-BUS into ACOR, and initiates a "low-request" command. When "selected", the ACOR register is output to the MCU bus, and the command and module number ("TO" lines) are obtained from the TO register and MOP register.

CRL — Enables the "STORE" field bus options (as above) to load TO register and MOP register from the U-BUS (CPU freezes until any pending MCU requests are completed). The registers are loaded as follows:

```
MOP(Ø:1) := UBUS(10:11)
 TO(2:4) := UBUS(13:15)
```

MOP register then contains a "command" (defined by the user) for the module whose address is contained in TO register.

DATA — Enables the "STORE" field bus options (as above) to store the U-BUS into DCOR, and initiates a "high-request" command.

DPOP — Same as "DATA" above, and in addition pops the stack (see "POP" in "SPEC" field).

DB — Same as "ABS" above, except specifies the DB-Bank register. This bank register is used with DB-relative addressing.

NIR — Enables the "STORE" field bus options (as above) to store the U-BUS into DCOR, and initiates a "high-request". On the following "select" cycle, DCOR is read onto the MCU bus and is then stored into the CPU "NIR" register (Next Instruction Register).

OPND — Same as "NIR" above, except the MCU bus is stored into the CPU "OPND" register (Operand Register).

PB — Same as "ABS" above, except specifies the PB-Bank register. This bank register is used with PB-relative addressing.

RND — Enables the "STORE" field bus options (as above) to load ACOR from the U-BUS, and initiates a "low-request" command. The DB-Bank register is used to generate the module number. This is used to initiate a data fetch from memory. The returned data is loaded into the NIR register.

RNP — Same as "RND" above, except the PB-Bank register is used to generate the module number.

RNS — Same as "RND" above, except the Stack-Bank register is used to generate the module number.

ROA — Same as "RND" above except:

1. The ABS-Bank register is used to generate the module number.
2. The data is returned to the OPND register.

ROD — Same as "RND" above except:

1. The DB-Bank register is used to generate the module number.
2. The data is returned to the OPND register.

ROND — Same as "RND" above, except the data is returned to both the NIR and OPND registers.

RONP — Same as "RND" above except:

1. The PB-Bank register is used to generate the module number.
2. The data is returned to both the NIR and OPND registers.

RONS — Same as "RND" above except:

1. The Stack-Bank register is used to generate the module number.
2. The data is returned to both the NIR and OPND registers.

ROP — Same as "RND" above except:

1. The PB-Bank register is used to generate the module number.
2. The data is returned to the OPND register.

ROS — Same as "RND" above except:

1. The Stack-Bank register is used to generate the module number.
2. The data is returned to the OPND register.

ROSA — Same as "RND" above except:

1. The ABS-Bank register is used to generate the module number.
2. The data is returned to the OPND register.
3. The word addressed is set to all 1's in memory.

ROSD — Same as "RND" above except:

1. The data is returned to the OPND register.
2. The word addressed is set to all 1's in memory.

S — Same as "ABS" above except specifies the Stack-Bank register. This bank register is used with DL, Q, or S-relative addressing.

WRA — Enables the "STORE" field bus options (as above) to load ACOR from the U-BUS, and initiates a "low-request" command. The ABS-Bank register is used to generate the module number. This is used to initiate a data store into memory. On the "select" cycle, the memory module addressed interprets the data on the MCU bus as an address, and goes "BUSY". It stays busy until it receives the data to be stored (normally sent on the following cycle with a microcode BUS DATA instruction) and completes its "write" cycle, or until its timer runs down.

WRD — Same as "WRA" above except the DB-Bank register is used to generate the module number.

WRS — Same as "WRA" above except the Stack-Bank register is used to generate the module number.

NOTE: ACOR and DCOR refer to the "Address CPU Output Register" and "Data CPU Output Register" respectively.

ERROR MESSAGES

1. INVALID CONTROL OPTION
2. INVALID ADDR OR ROM K EXPRESSION
3. UNDEFINED LABEL
4. RBUS,SHIFT FIELDS INVALID WITH ROM FNS
5.
6.
7.
8.
9.
10. SKIP FIELD INVALID WITH STORE RAR
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23. INVALID RBUS OPTION
24. INVALID SBUS OPTION
25. INVALID FUNC OPTION
26. INVALID SHFT OPTION
27. INVALID STOR OPTION
28. INVALID SKIP OPTION
29.
30. INFINITE REPEAT LOOP
31. INVALID SPEC OPTION
32. INVALID SPEC/MCU OPTION
33.
34.
35.
36.
37. INVALID REPN CONSTANT
38.
39. DUPLICATE LABEL
40. FORMAT ERROR
41. INVALID MCU OPTION
42. CLSR CONFLICTS WITH NEXT STACK PREAJUST
43.
44.
45.

WARNING MESSAGES

1.
2. RBR CONFLICTS WITH PREFETCH ON INSTR ENTRY
3. RBUS MAY BE FORCED ON FOLLOWING LINE
4. SBUS MAY BE FORCED ON FOLLOWING LINE
5. CCA,CCZ SET ON TBUS
6. PRECEDING TOS STORE NAME AFFECTED BY MREG OR QDWN
7. TOS LOAD NAME AFFECTED BY PRECEDING MREG OR QUP
8. TOS LOAD NAME IS OLD NAME BEFORE PRECEDING PUSH, POP OR INCN
9.
10.
11.
12. ZERO,NZRO,NSME SKIP TESTS MADE ON T-BUS
13.
14.
15. CLIB MAY BE TOO CLOSE TO JLUI
16. BOUNDS TEST WITH RRZ,RLZ,LRZ,LLZ DOES A CAD
17. UBUS ON SBUS OR SR1 MISSING FROM MPAD
18. UBUS ON RBUS OR SL1 MISSING FROM DVSB
19. SL1 OR SR1 AS APPROPRIATE MISSING
20. FUNCTION AND STORE SP3 MAY CONFLICT
21. FUNCTION AND STORE SP1 MAY CONFLICT
22. RBUS ON RBUS INHIBITS CONTENTS CHANGING
23. SBUS ON SBUS INHIBITS CONTENTS CHANGING
24. SKIP CONDITION MISSING FROM JMP,JSB
25. STORE OR STORE-INCT CTR CONFLICT
26. STORE/SET/CLR CC,OVFL,CRRY STA BITS CONFLICT
27.
28. SR CHANGE CONFLICT
29. NAMER CHANGE OR TNAME CONFLICT
30. OLD PB-BNK USED FOR NEXT PREFETCH

INTERRUPTS

The following is a brief explanation of the hardware interrupt information available to the microprocessor, and the hardware dependent sequences that the microprocessor must execute to handle interrupts correctly. Interrupts are detected via the interrupt status registers CPX1 and CPX2. CPX1 contains all run state interrupts and status information, that is, those that occur while the CPU is executing instructions, and CPX2 contains all control panel interrupts in addition to halt-made status information. The special field option CCPX is used to control the information in the interrupt registers and hardware dependent sequences.

Note that all interrupt bits in both CPX1 and CPX2 will cause a hardware jump to ROM address 3 in a NEXT skip field option is executed, and also will cause the TEST skip condition to be true.

Control of the interrupt bits is accomplished by the run-halt state. In run state, all interrupt bits in CPX1 are allowed, while all interrupt bits in CPX2 are held off. In halt state, front panel interrupts are allowed, while all interrupt bits in CPX1 except power-fail (CPX1(9)) are held off. Interrupt bits in CPX1 (except CPX1(Ø), the integer overflow bit) are cleared by executing a CCPX "SPEC" field option with a field decoded from UBUS(4:7).

Note that "clearing" the power fail interrupt inhibits all other interrupt bits in both CPX1 and CPX2. All interrupt bits in CPX2 are cleared by executing a CCPX with bit 15 of the U-BUS being a 1.* Note that run, execute switches, and single instruction interrupts must be cleared before the execution of a NEXT. A complete description of CPX1, CPX2 and the spec field function CCPX is given on the following pages.

*Interrupt bits include CPX2(0:8).

CPX1, CPX2 Bit Assignments and SPEC field CCPX Option

BIT	CPX1	CCPX	FIELD	CPX2
0	Integer OVFL	Halt	NOP	Run Sw.
1	Bounds Viol.	Run	Clr. BNDV	Dump Sw.
2	Illegal Addr.	Sys. Halt	Clr. Ill. Addr.	Load Sw.
3	CPU timer		Clr. CPU Timer	Load Reg.
4	Sys. P.E.	MSB	Clr. Sys. P.E.	Load Addr.
5	Addr. P.E.		Clr. Addr. P.E.	Load Mem.
6	Data P.E.	Field Code	Clr. Data P.E.	Disp. Mem.
7	*Module Intrp.	LSB	Clr. Mod Intrp.	Sing Instr.
8	** Ext. Intrp.	Diag. NIRTOCIR	Clr. Ext. Intrp.	Exec. Sw.
9	Power Fail		Pwf Turn-off Int.	Incr. Addr.
10		Diag. Set CPX1(1:8)		Decr. Addr.
11	ICS Flag	Clr. ICS Flag		
12	Disp. Flag	Clr. Disp. Flag		
13	Emulator			Inh. PFARS
14	I/O Timer	Diag. Freeze	Rev. Sys. Parity	Sys. Halt
15	Option Present	Clr. Panel FF's.	Rev. MCUD Parity	Run FF.

*Interrupt is enabled by STA(1)

**Interrutp Poll is enabled by STA(1)

BIT

Ø — Integer Overflow. This is the logical "AND" of STA(2) (user traps bit in STATUS reg.) and STA(4) (overflow bit in STATUS reg.). It allows an interrupt only if both are set. It can only be cleared by clearing either (or both) bits in the STATUS reg.

1 — Bounds Viol. This bit is set whenever an attempt is made to address outside the users assigned environment (i.e. USERMODE·[(E<DL or E>S) or (E<PB or E>PL)]. See ERS for complete bounds check information.

2 — Illegal Addr. This bit is set when an attempt is made to address non-existent memory (i.e. an address larger than the amount of memory that is physically in the system). The bus transmission for this attempt is inhibited.

3 — CPU Timer. This bit is set if the CPU does not receive a response from a module it had previously addressed within 4.6 ms. It also forces the CPU out of any freeze state it might be in so that it can complete the instruction and service the interrupt. The result of the instruction in this case is normally garbage. (This is also referred to as a "non-responding module" interrupt.)

4 — Sys. Parity Error. This bit is set if a parity error is detected on the 8-bit system information (TO, FROM, COMMAND) on a CPU to memory or memory to CPU transmission.

5 — Addr. Parity Error. This bit is set if a memory module detects a parity error on an address transmitted from the CPU.

6 — Data Parity Error. This bit is set if the CPU detects a parity error on the data transmitted from a memory module. Note that if memory receives data with bad parity (on a write cycle), it will store the information as received. No error information is generated.

BIT

7 Module Interrupt. This bit is set if the CPU receives a command, with good system parity, and it is not expecting it. Note that, as well as detecting a possible error, it can also be used as a "semaphore" between the CPU and another module (e.g. a 2nd CPU) for information swapping (i.e. one CPU can send a command to the 2nd CPU saying, in effect "look in your mailbox (a known core location) for the information I am transmitting". This search would be done in the module interrupt routine.)

8 External Interrupt. This bit is set when a device (not masked off) is requesting service.

9 Power Fail. This bit is set when a power failure is detected.

11 ICS Flag. Set=1 when the machine is executing on the "Interrupt Control Stack".

12 DISP. Flag. Set=1 when the machine is in the dispatcher. Since the dispatcher executes on the ICS, CPX1(11) will also be set during this time.

13 Emulator. Set=1 by a switch on the ROM board when the /20 emulator μ-code is being executed. Useful for "DPAN".

14 I/O Timer. Set=1 if an I/O device does not respond to a "Service Out" request or "data poll" within 3 μsec. This does not generate an interrupt; instead it is tested in the μ-code which executes the I/O instructions, and its state is indicated in the STATUS reg "condition code" upon completion of the I/O instruction. This bit must be tested following the issuance of any I/O command (i.e. executing a "STORE" field IOA). It is cleared on the cycle following the reading of CPX1 (following "IOA") which allows testing the bit.

15 Option Present. Used to indicate whether or not a given instruction set option is present (using % 0204XX as the entry opcode). It is tested by μ-code.

10 = Ø (Unused).

BIT

Ø RUN SW. Set = 1 when the "RUN/HALT" switch is depressed. This, in conjunction with the state of CPX2(15), is used to put the machine in the "RUN" or "HALT" mode.

1 DUMP SW. Set = 1 when the "SYSTEM DUMP" switch is depressed.

2 LOAD SW. Set = 1 when the "COLD LOAD" switch is depressed.

3 LD REG. Set = 1 when the "LOAD REG." switch is depressed.

4 LD ADDR. Set = 1 when the "LOAD ADDR" switch is depressed.

5 LD MEM. Set = 1 when the "LOAD MEM" switch is depressed.

6 DISP. MEM. Set = 1 when the "DISPLAY MEMORY" switch is depressed.

7 SINGLE INSTR. Set = 1 when the "SINGLE INSTRUCTION" switch is depressed.

8 EXECUTE SW. Set = 1 when the "EXECUTE SWITCH REG." switch is depressed. The software instruction contained in the "SYSTEM SWITCH REGISTER" is executed.

NOTE: The above bits (CPX2(Ø:8)) are defined to be the "halt mode interrupts", and are enabled only when the machine is in "HALT" mode. They force the machine to jump to the μ-code interrupt handler where they are scanned in sequence to determine the action to be taken. When a bit is found = 1, a jump to a μ-code routine is taken, the interrupt is serviced, and (except for CPX2(Ø)) the Machine returns to the "HALT" mode - the "RUN" FF is not turned on. As soon as the bit causing the interrupt is detected, it is cleared by an "INC CCPX" μ-instruction to prevent further interrupts.

The following bits in CPX2 contain miscellaneous panel information used only by the μ-code.

?IT

9 INC. ADDR. Set = 1 if the Control Panel Memory Address INCREMENT switch and ENABLE switch are both on. This bit is checked by the μ-code to decide whether or not to increment the address following a "load" or "display" memory.

10 DEC. ADDR. Same as bit 9 above except the switch must be in the "DECREMENT" position, and is used to test whether or not to decrement the memory address.

13 INH. AUTO-RES. Set = 1 if the Control Panel Auto-Restart switch is set to the "INHIBIT" position.

14 SYSTEM HALT. Set = 1 by the μ-code if a "SYSTEM HALT" condition is detected.

15 RUN FF. Set = 1 by the μ-code when the machine is put in the "RUN" mode.

11,12 = Ø

SPECIAL Field Option "CCPX" Definitions

The following action takes place when the bit referred to exists on the UBUS in conjunction with the μ-op "CCPX" in the SPEC. field.

BIT

Ø — HALT. Clears the "RUN" FF (i.e. go to HALT).

1 — RUN. Sets the "RUN" FF.

2 — SYSTEM HALT. Sets the "SYSTEM HALT" FF. This FF can only be cleared by "PON" or "SYSTEM RESET". When set, all interrupts are inhibited except "SYS DUMP", "LOAD REG", "LOAD/DISPLAY MEM", and "PWR FAIL".

3 — Unused.

4:7 — CLEAR CPX1. This field is decoded into 1 of 16, ANDed with CCPX, and used to clear the appropriate bit in CPX1. These bits currently include 1:9, 11, 12, 14, 15. (Bit Ø is cleared by clearing OVFLO or USER TRAPS bit in the STATUS REG.) Bit 9 is not acutally cleared by this field; instead, a FF is set which inhibits any further interrupts of any type. This FF is cleared by "PON" or "SYSTEM RESET". The field decode/CPX1 bit correspondence is shown below:

UBUS(4:7) = 0000	NOP	
= 0001	clears CPX1	(1)
= 0010	"	(2)
= 0011	"	(3)
= 0100	"	(4)
= 0101	"	(5)
= 0110	"	(6)
= 0111	"	(7)
= 1000	"	(8)
= 1001	"	(9)
= 1010	NOP	
= 1011	NOP	
= 1100	NOP	
= 1101	NOP	
= 1110	See "A"	
= 1111	See "B"	

BIT

4;7 (Cont.)

A. Sets a FF which complements the system data parity bit. Remains complemented until a second (UBUS(4:7)=1110)·CCPX is executed. Used for diagnostic purposes only.

B. Same as "A" except uses (UBUS(4:7)=1111)·CCPX, and is used to complement the MCU data parity bit. Can be used to test either address or data parity.

8 Diag. NIRTOCIR. Causes the contents of NIR to be loaded into CIR. Used mainly for μ-diagnostics. Note that, due to the "pipe" structure of the CPU, you must delay 1 clock before reading CIR to the S-BUS. This also resets the "CLIB" FF.

9 Unassigned.

10 Diag. SET CPX1(1:8). Sets the interrupt FF's in CPX1 corresponding to bits 1:8. This is used for μ-diagnostics.

11 CLEAR ICS FLAG. Clears the "INTERRUPT CONTROL STACK" flag FF.

12 CLEAR DISP. FLAG. Clears the "DISPATCHER" flag FF.

13 Unassigned.

14 Diag. FREEZE. Sets the "FREEZE" FF which turns off the clock to the CPU. This forces the CPU to stop execution upon completion of the μ-instruction containing UBUS(14)·CCPX. This FF is cleared by the Control Panel "RAR BREAKPOINT HALT/FREEZE-EXIT" switch. This function is used only for μ-diagnostics.

15 CLR PANEL FF's. Sends a reset (clear) signal to CPX2(Ø:8). This is, in effect, a master clear for these interrupts.

MICRO-PROGRAMMING NOTES

The following is a random collection of notes which attempt to explain some obscure cases in micro-programming.

1. The micro-assembler does not recognize the option "NOP" in any of the fields, and hence will generate an error message. To "NOP" a field, it must be left blank (note that the "FUNCTION" field may not by "NOPed" - use ADD for this case).

2. BNDT, UBNT require one cycle only to execute. If the trap is taken (a fault detected), two overhead cycles are required (Freeze, NOP2) before the execution of the micro-instruction at ROM ADDRESS 2.

3. JLUI and RSB can be executed from RANK1 if the line of microcode in RANK2
   a) is cancelled by NOP2
   b) contains a ROM function
   c) contains a NOP skip test
   d) contains a non-data dependent skip test (options 14-27, 32-34) which is not met

   or if a previous JMP/JSB-UNC has just been taken from RANK1.

   These all result in a zero-overhead JLUI or RSB.

4. a) JMP/JSB-UNC can be executed from RANK1 if the line of microcode in RANK2
      1) is cancelled by NOP2
      2) contains a ROM function
      3) contains a NOP skip test
      4) contains a non-data-dependent skip test (options 14-27, 32-34) which is not met

      or if a previous JMP/JSB-UNC, JLUI, or RSB has just been taken from RANK1.

Otherwise JMP/JSB-UNC is executed from RANK2. If the JMP/JSB is executed from RANK1, there are no overhead clocks required. The micro-instruction jumped to will execute on the clock following the execution of the JMP/JSB micro-instruction. Note that, although the JMP portion of the micro-instruction may execute in RANK1, the remainder of the instruction executes in RANK2. Hence the timing for micro-code of the form

```
          RA  RB   ADD   --   SP2  --  --
          --  RC   JMP  TARG  SP3  --  UNC
                    :
                    :
                    :
    TARG  --  SP2  INC   --   SP1  --  --
```

is as follows:

Clock 1	Clock 2	Clock 3
In RANK2	In RANK2	In RANK2
RA + RB → SP2	RC → SP3	SP2 + 1 → SP1
In RANK1	In RANK1	In RANK1
TARG + 1 → RAR	(TARG) = SP2 INC SP1	(TARG + 1)
(TARG) → RANK1 Input		

4. b) JMP/JSB-UNCs which are executed from RANK2 because none of the above fast-jump conditions were present, and conditional JMP/JSB's which are always executed from RANK2 behave as follows:
   1) NOT TAKEN - next line in sequence executed on next clock
   2) NON-DATA-DEP TAKEN - one overhead clock required (NOP2) before target line executed
   3) DATA-DEP TAKEN - two overhead clocks required (FREEZE, NOP2) before target line executed

   Execution of JMP/JSB (or RSB, JLUI) in RANK2 inhibit any fast jump execution fram RANK1. Hence, if there are two consecutive lines of micro-code containing JMP, and the JMP in the first line is taken from RANK2, the JMP in RANK1 will be ignored.

If NOP2 is set, any inhibits from this rank that might have held off fast jumps from RANK1 are removed. This would allow code such as the following to execute with the timing shown:

```
        --  RA  JMP  X  SP3  --  F1   All conditional JMP/JSB's
                  :                    execute in RANK2
                  :
                  :
    X   --  RB  JSB  Y  SP2  --  UNC
```

timing

Clock 1	Clock 2	Clock 3
JMP instr in RANK2	NOP2	JSB instr in RANK2
anything in RANK1	JSB instr in RANK1	RB → SP2
RA → SP3	Y + 1 → RAR	(Y) in RANK1
Assume F1 = 1	(Y) → RANK1	etc.
X + 1 → RAR		
(X) → RANK1		

5. The following describes how the CPU will behave for various cases of JMP/JSB's with possible operand freezes. Note that this is not necessarily an all-inclusive list of cases.

1)
```
    RA  RB    ADD   --   RC   --  F1
    --  UBUS  JMP   X    SP3  --  UNC
    --  OPND  ADD   --   SP1  --  --
                :
                :
X   --  OPND  ADD   SWAB SP1
```

A) F1 = Ø. "JMP X - UNC" will execute from RANK1. Hence the third line of micro-code will never be seen. However, when "JMP X - UNC" is in RANK2 to complete its execution, (X) is in RANK1. An OPND freeze (if required) would now occur.

B) F1 = 1. Now the JMP will not be taken. Instead, when "JMP X - UNC" is loaded into RANK2, NOP2 is also set. However, the third line of micro-code (containing "OPND ADD - SP1") is also loaded into RANK1 on this clock, and an OPND freeze due to it could occur.

2)
```
    --  RA    JMP  X   SP3  --  ZERO
    --  OPND  ADD  --  SP1  --  --
                 :
                 :
X       OPND  ADD  --  SPØ  --  --
```

This "JMP X" will always execute from RANK2 since it is a data-dependent conditional jump.

A) (RA) ≠ Ø. No jump taken. The line of micro-code in RANK1 containing "OPND ADD - SP1" would cause an OPND freeze, if required.

B) (RA) = Ø. Jump is taken. A one-clock freeze is forced by the data-dependent condition in order to get X + 1 → RAR and (X) → RANK1 (→ may be read "to the input of"). This over-rides any previous RANK1 freeze (e.g. OPND). This is followed by NOP2 to fill the pipe, at which time RANK1 freezes are re-enabled. Note that this can, in effect, stretch out the NOP2 cycle due to a freeze. The timing sequence is shown below:

	no clock	clock
	Time Period 1	Time Period 2	Time Period 3
	JMP X - ZERO in RANK2	Freeze period	NOP2
	UBUS = Ø = freeze clock	X + 1 → RAR	(X) in RANK1
		(X) → RANK1	(freeze if req'd)
		over-ride OPND freeze	
		RA → SP3	

3)
```
    --  RA    ADD  --  --  --  ZERO
    --  --    JMP  x   --  --  UNC
    --  OPND  ADD  --  SP1 --  --
                 :
                 :
X   --  OPND  ADD  --  SP3 --  --
```

Again, the "JMP X - UNC" will execute in RANK2 since it is preceded by a data-dependent skip condition. The two cases here are

A) (RA) = Ø. The "ZERO" test will set NOP2, forcing the "JMP X - UNC" in RANK2 to be NOPed. The "OPND ADD - SP1" micro-instruction in RANK1 can force an OPND freeze (if required), in effect extending the NOP2 cycle.

B) (RA) ≠ Ø. The micro-instruction "JMP X - UNC" will now execute from RANK2. "OPND ADD - SP1" in RANK1 may try to force an OPND freeze. However the act of executing a JMP will over-ride this freeze for one cycle. When RANK1 is loaded with (X), any freeze condition implicit in this instruction is enabled. The timing for this sequence is shown below:

Clock 1	Clock 2	Clock 3
JMP X - UNC in RANK2 X + 1 → RAR (X) → RANK1 OPND ADD - SP1 in /RANK1 OPND freeze over-ridden	NOP2 OPND ADD - SP1 in RANK2 ignored OPND ADD - SP3 in RANK1 - possible freeze.	Possible freeze period. If not, OPND ADD - SP3 in RANK2 OPND → SP3

6. Some confusion may exist concerning the state of the RBUS reg., SBUS reg., and UBUS following different kinds of JMP/JSB's. The following examples may help to clear this up. As a point of interest, it may be noted that JMP/JSB and ROM functions (ROM, ROMN, etc.) are always decoded in RANK1 in order to determine what to do with the RBUS reg. On the clock edge where the JMP/JSB micro-instruction is transferred from RANK1 to RANK2, the RBUS register is loaded with all Ø's, and for the ROM function it is loaded with the ROM constant. The normal R-field decode is inhibited for these cases. As an additional note of interest, it may be seen that it is possible for the four least significant bits of the JMP/JSB target or ROM constant to appear to be the R-field decode of "RBUS". This would tend to inhibit clocking the RBUS reg. Special hardware has been put into the CPU (the RFINH signal) to prevent this, thus insuring the RBUS register will clock.

1) RANK1 Jump Taken. Assume code of the form

```
--  SPØ  ADD  --  SP2  --  --
--  SP1  JMP  X   --   --  UNC
RA  RB   ADD  --  SP3  --  --
```

then

A) X RBUS SP3 ADD ... ; UBUS ← Ø + (SP3)
B) X UBUS SP3 ADD ... ; UBUS ← (SP1) + (SP3)
C) X - UBUS ADD ... ; UBUS ← (SP1)
D) X - SBUS ADD ... ; UBUS ← (SP1)
E) X RA SBUS ADD ... ; UBUS ← (RA) + (SP1)

2) RANK2 Jump Taken. Assume code of the form

```
RA  RB   ADD  --  SP1  --  --
--  SP3  JMP  X   --   --  POS
RC  RD   ADD  --  --   --  --
```

then

A) X - UBUS ADD ... ; UBUS ← (RC) + (RD)
B) X RBUS - ADD ... ; UBUS ← (RC)
C) X - SBUS ADD ... ; UBUS ← (RD)
D) X UBUS SP2 ADD ... ; UBUS ← (RC) + (RD) + (SP2)

3) RANK2 Jump Not Taken. Assume code of the form

```
RA  RB   ADD  --  SP1  --  ZERO    assume UBUS = Ø
--  SP3  JMP  X   --   --  UNC     not taken
```

then if the next sequential line is:

A) RBUS - ADD ... ; UBUS ← Ø
B) - SBUS ADD ... ; UBUS ← (SP3)
C) UBUS UBUS ADD ... ; UBUS ← (SP3) + (SP3)
D) RC RD ADD ... ; UBUS ← (RC) + (RD)

7. In order to simplify the micro-code in the memory-reference address calculations, a FF has been put in the CPU to provide automatic bank register selection between the DB bank register and the stack bank register. It works as follows:

   The micro-code always specifies the DB bank for DQS address calculations. If, for mem. ref. instructions (and not loop control (TBA, etc.)), Q- or S-rel. addressing is specified, the FF is set. When the memory reference is made, the bank reg. pointed to by this FF is appended to the leading bits of the calculated address to form the 18 bit address.

The FF is cleared (so that the MCU option "ROD" always points to the DB bank) by the spec. field option "CLIB", or by "NEXT" or "System Reset".

This is a special-purpose FF which can only be set through sub-ops 04-17, and hence is of no use to the general-purpose micro-programmer.

8. Do not attempt to execute a memory operation between issuing a "BUS CRL" and "BUS CMD". The "TO" and "OPERATION" information for the "CRL" will be lost if this is done. Refer to HP 3000/20 Rev. E micro-code for the CMD instruction (@2356 - @2362) for an example of code that will be invalid on the HP 3000/30. (Note: a CRL must be issued prior to each CMD to insure correct MCU operation.)

9. When storing data into memory, the data is normally sent on the line of micro-code immediately following the address transmission. Under no circumstances should more than one line of micro-code be inserted between the address transmission and the data transmission (I/O bandwidth could be affected). Also, this line should not contain a BUS-OP. This would cause the "TO" information from the address transmission to be lost.

10. When a line of micro-code is skipped, the function field is changed to "ADD", the shift, store, spec., and skip fields are NOP'ed, and UBUS ← (RBUS) + (SBUS). If the function field of the skipped micro-instruction contains ROM, ROMI, ROMN, or ROMX, the RBUS reg. will be loaded with Ø's instead of the ROM constant.

11. Interrupts are checked on the clock cycle following the execution of "NEXT" in RANK2. Hence, a line of micro-code such as

    * * * * * SOV NEXT (* = valid μ-op)

    would (if the user traps bit in the status word were set) cause an interrupt. Replacing "SOV" with "CLO" would prevent an OVFLO interrupt.

    The External Interrupts Enable/Disable bit (STA(1)) is handled differently. See the HP 3000/30 ERS for an explanation of its effect.

    As a result of the above, ADDO, CADO, INCO, and SUBO are now one cycle operations.

12. The line of micro-code pointed to by an L.U.T. entry may not contain "JLUI" (this is normally the first line of a micro-program). This is due to hardware limitations of the "NEXT" sequence.

13. Subroutines may be placed in-line in critical spots. "RSB" is treated as a "NOP" unless a "JSB" has been previously executed. Subroutines may be exited by "RSB" or "NEXT".

14. "QASL", "QASR" are normally executed in a "REPEAT" loop where any register-handling anomalies are handled by the pipe. It is possible, however, to do a single quadruple-reg. shift outside a "REPEAT" loop. For example, assume it is desired to do a one-bit "QASL". Let

    RA ← hi-bits
    RB ← next most bits
    RC ← next least bits
    RD ← lo-bits

    The following code would execute the shift:

```
RB   --  ADD  --  SP3  --  --    SP3 ← next-most bits
RC   --  ADD  --  SP1  --  --    SP1 ← next-least bits
RD   RA  QASL SL1 SPØ  --  --    RBUS← lo-bits
(1)  --  ADD  --  SP2  --  --    SBUS← hi-bits
```

    Upon completion of the code, SPØ ← hi-bits, SP3 ← next-most bits, SP1 ← next-least bits, and SP2 ← lo-bits. Note that (1) in the third line of code (left blank) is an implied "RBUS".

15. One-line subroutines are not allowed.

16. It is legal to do "STORE" and "NEXT" on the same line of micro-code; however, interrupt information based on the state of the "USER TRAPS BIT" and "OVFLO" (STA(2) and STA(4)) will be from the old state of the STATUS reg. - not the new. It is <u>not</u> legal to do this if STA(1) (External Interrupt bit) is changed from 1 to Ø. Since IPOLL can take up to 925 ns to detect the interrupting device, it could be possible to have an Ext. Int. occur on the following instruction even though STA(1) had been turned off in the current instruction. If STA(1) is to be disabled, it should be done 1 µsec (6 clocks) <u>before</u> executing "NEXT".

    Also, it is not legal if the "Right Stack-op Pending" bit (STA(3)) can be changed. This affects the "NEXT" sequencer. If this bit can be changed by storing status, then the store must be done at least <u>2</u> lines preceding "NEXT".

17. If a line of μ-code contains both a ROM function (ROM, ROMI, ROMN, ROMX) and a BUS-OP (BUS, BSPØ, BSP1) in the store field, an implied DATA will be issued to the MCU. This is convenient for μ-diagnostics for storing programs in memory. For example, if SPØ contains an address, then

```
SPØ  —  ADD  —  BUS  WRA  —
 —   —  ROM  —  BUS  101000
```

is sufficient to store the constant %101000 at (SPØ) in memory.

18. It is legal to execute "Store P" and "NEXT" on the same line of μ-code - e.g.

```
SPØ  —  INC  —  P  —  NEXT
```

If this case is detected, the address used for the pre-fetch in NEXT is taken from the U-BUS rather than the P-reg.

19. The Subroutine Flag FF is set by execution of "JSB" in the ROM Function Field and "condition met" in the Skip field, and is cleared by "RSB", "NEXT", PON, system reset, or detection of a bounds violation (using BNDV or UBNT).

20. Execution of the line of μ-code

```
—  RBR  ADD  —  BUS  DATA  —
```

results in the DB-Bank being loaded into DCOR. This is accomplished by pre-decoding RBR·DATA in RANK1 and loading DB-BANK into the RBUS register.

21. The sequences

```
*   *   *   *   *   *   NEXT
*  RBR  *   *   *   *    *
```

or

```
*   *   *   *  SBR  *   NEXT
```

are not allowed (* = valid μ-op. It could foul up the bank register select (PB-Bank) for the NEXT pre-fetch.

22. The first line of a μ-program executed following the "NEXT" sequence cannot contain "RBR" or "JLUI". The "NEXT" sequence will not execute properly if this is the case.

23. "CLIB" must be executed at least 2 lines before "JLUI" (since "JLUI" can execute from RANK1).

24. Assume a typical sequence of μ-code as shown below:

```
*     *       ADD     *     BUS     ROD     *     * = any valid μ-op.
                :
*     *       ADD     *     BUS     WRA     *
*     OPND    ADD     *     BUS     DATA    *
```

Do not insert a line of μ-code between the lines containing "WRA" and "DATA". "OPND" is used to freeze (hold-off) the write command, if necessary, due to the preceding data fetch.

25. Following are some restrictions on the function field μ-op "DCAD";
   A. Do not use with store field ops "BUS, BSPØ, BSP1, or SBR".
   B. Do not use with skip field ops "POS, NEG, BIT6, BIT8".
   C. Do not use with spec field op "CCB".

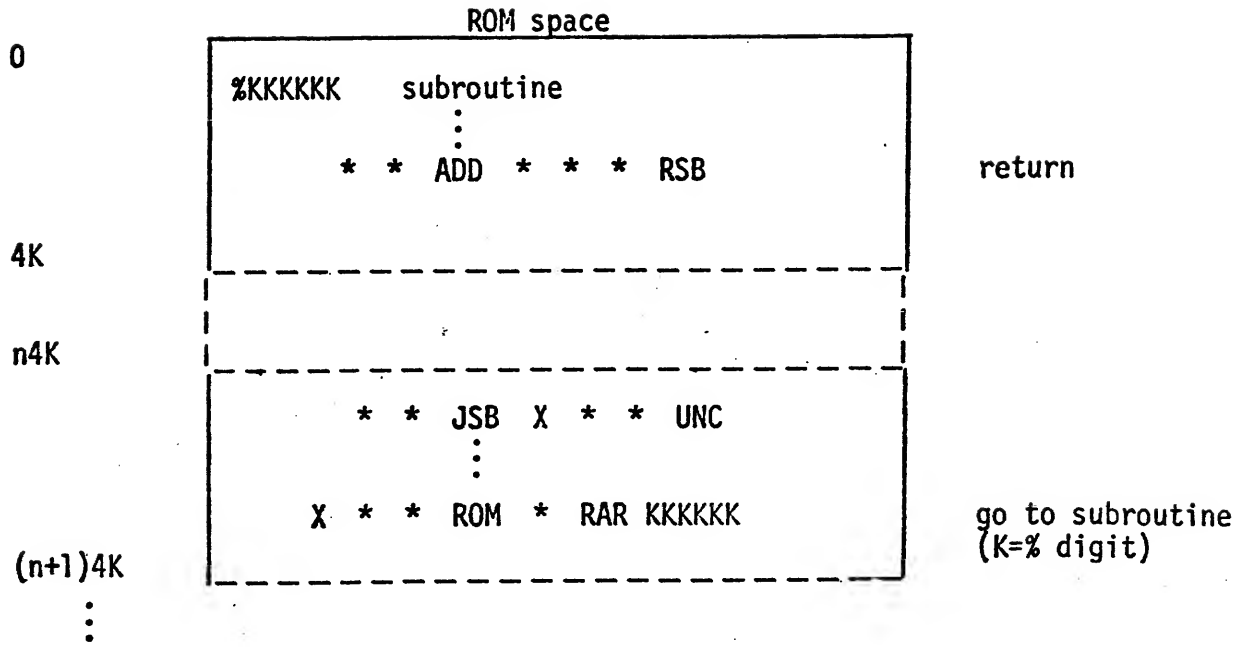
   In general these are timing constraints due to the 2nd level of addition in this function.

26. The timing sequence for the store field op "RAR" is shown below:

execute STORE RAR in RANK2	JMP FRZ	NOP2	execute μ-inst. at new addr.
Data → UBUS	UBUS → VBUS →ROM→RANK1	RANK1→RANK2	

27. Due to the 12-bit address space in the μ-word format, JMP's & JSB's are limited to a 4K range. On JSB, however, the entire 16-bit RAR is saved in the SAVE reg., and restored on RSB. Therefore if you wish to use a subroutine in a 4K bank (of ROM/RAM) other than the one you are currently executing in, it may be accomplished as shown in the following example (note: this is intended for future applications only): Continued on next page.

27. (Cont.)


ROM space
0
%KKKKKK subroutine
* * ADD * * * RSB
return
4K
n4K
* * JSB X * * * UNC
X * * ROM * RAR KKKKKK
go to subroutine
(K=% digit)
(n+1)4K